BOUNDARY-FITTED CURVILINEAR COORDINATE SYSTEMS FOR SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS ON FIELDS CONTAINING ANY NUMBER OF ARBITRARY TWO-DIMENSIONAL BODIES

Joe F. Thompson, Frank C. Thames, and C. Wayne Mastin

Prepared by
MISSISSIPPI STATE UNIVERSITY
Mississippi State, Miss. 39762
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JULY 1977
# Boundary-Fitted Curvilinear Coordinate Systems For Solution of Partial Differential Equations on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies

Joe F. Thompson, Frank C. Thames, and C. Wayne Mastin

Mississippi State University
Mississippi State, Mississippi 39762

National Aeronautics & Space Administration
Washington, DC 20546

Contractor Report

Ruby Davis of the Theoretical Aerodynamics Branch of Langley Research Center converted the UNIVAC 1106 code generated at Mississippi State University to the CDC 6000 Series code at Langley Research Center. Percy J. Bobbitt was the Langley Technical Monitor. Final report.

A method for automatic numerical generation of a general curvilinear coordinate system with coordinate lines coincident with all boundaries of a general multi-connected two-dimensional region containing any number of arbitrarily shaped bodies is presented. No restrictions are placed on the shape of the boundaries, which may even be time-dependent, and the approach is not restricted in principle to two dimensions. With this procedure the numerical solution of a partial differential system may be done on a fixed rectangular field with a square mesh with no interpolation required regardless of the shape of the physical boundaries, regardless of the spacing of the curvilinear coordinate lines in the physical field, and regardless of the movement of the coordinate system in the physical plane. A number of examples of coordinate systems and application thereof to the solution of partial differential equations are given. The FORTRAN computer program and instructions for use are included.

### Key Words (Suggested by Author(s))
- Coordinate transformation
- Multielement airfoils
- Navier-Stokes

### Distribution Statement
- Unclassified - Unlimited
- Subject Category 02

### Security Classification
- Unclassified

**For sale by the National Technical Information Service, Springfield, Virginia 22161**
TABLE OF CONTENTS

I. Introduction .................................. 1

II. Mathematical Development ..................... 7
   Preliminaries ................................ 7
   Doubly-Connected Region. .................... 10
   Multiply-Connected Region. .................. 13
   Simply-Connected Region. .................... 15
   Coordinate System Control .................... 15
   Time-Dependent Coordinate Systems. .......... 19

III. Numerical Solution .......................... 21
   Difference Equations ........................ 21
   Multiple-Body Fields ........................ 23
   Multiple-Body Segment Arrangements .......... 24
   Initial Guess ............................... 26
   Convergence Acceleration .................... 30
   Optimum Acceleration Parameters ............. 33

IV. Examples of Coordinate Systems ............... 36
   Coordinate System Control .................... 36
   Various Body Shapes. ........................ 37

V. Examples of Application to Partial Differential Equations .... 38
   Potential Flow ............................. 38
   Viscous Flow ............................... 42
      Velocity-Pressure Formulation .......... 44
      Pressure Distribution and Force Coefficients. 45
   Loaded Plate .............................. 46

VI. Instructions for Use - Coordinate Program ...... 47
   Coordinate System .......................... 47
      Dimensions ................................ 48
      Input Parameters ......................... 48
      Field Size .............................. 48
      Plotting ................................ 48
      Initial Guess ............................ 49
      Body Contours ........................... 49
      Re-entrant Boundaries .................... 50
      Acceleration Parameters ................ 51
      Coordinate System Control ............... 51
      Convergence of Very Large Fields .......... 54
   Listing of Program TOMCAT ................... 56

iii
# TABLE OF CONTENTS, CONTINUED


<table>
<thead>
<tr>
<th>Sample Cases</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. Simply-Connected Region</td>
<td>94</td>
</tr>
<tr>
<td>Input</td>
<td>94</td>
</tr>
<tr>
<td>Output</td>
<td>96</td>
</tr>
<tr>
<td>Plot</td>
<td>101</td>
</tr>
<tr>
<td>2. Single-Body Field</td>
<td>102</td>
</tr>
<tr>
<td>Input</td>
<td>102</td>
</tr>
<tr>
<td>Output</td>
<td>103</td>
</tr>
<tr>
<td>Plot</td>
<td>109</td>
</tr>
<tr>
<td>3. Double-Body Field</td>
<td>110</td>
</tr>
<tr>
<td>Input</td>
<td>110</td>
</tr>
<tr>
<td>Output</td>
<td>112</td>
</tr>
<tr>
<td>Plot</td>
<td>122</td>
</tr>
</tbody>
</table>

VII. Instructions for Use - Factors Program............. 123

| Scale Factors                         | 123  |
| Listing of Program FATCAT             | 124  |
| Sample Case: Single-Body Field        | 140  |
| Input                                 | 140  |
| Output                                | 141  |

VIII. References......................................... 147

IX. Tables.................................................. 150

X. Figures................................................. 170

XI. Appendices............................................ A-1
BOUNDARY-FITTED CURVILINEAR COORDINATE SYSTEMS
FOR SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS
ON FIELDS CONTAINING ANY NUMBER OF ARBITRARY TWO-DIMENSIONAL BODIES

By Joe F. Thompson,† Frank C. Thames,* and C. Wayne Mastin§
Mississippi State University
Mississippi State, Mississippi 39762

Research Sponsored by NASA Langley Research Center, Grant NGR 25-001-055

SUMMARY

A method for automatic numerical generation of a general curvilinear coordinate system with coordinate lines coincident with all boundaries of a general multi-connected, two-dimensional region containing any number of arbitrarily shaped bodies is presented. No restrictions are placed on the shape of the boundaries, which may even be time-dependent, and the approach is not restricted in principle to two dimensions. With this procedure the numerical solution of a partial differential system may be done on a fixed rectangular field with a square mesh with no interpolation required regardless of the shape of the physical boundaries, regardless of the spacing of the curvilinear coordinate lines in the physical field, and regardless of the movement of the coordinate system in the physical plane. A number of examples of coordinate systems and application thereof to the solution of partial differential equations are given. The FORTRAN computer program and instructions for use are included.

I. INTRODUCTION

There arises in all fields concerned with the numerical solution of partial differential equations the need for accurate numerical representation of boundary conditions. Such representation is best accomplished when the boundary is such that it is coincident with some coordinate line, for then the boundary can be made to pass through the points of a finite difference grid constructed on the coordinate lines; hence the choice of cylindrical coordinates for circular boundaries, elliptic coordinates for

†Professor of Aerophysics and Aerospace Engineering, Ph.D.

*Engineering Specialist, Ph.D., Present Affiliation: LTV Aerospace Corporation, Dallas, Texas.

§Associate Professor of Mathematics, Ph.D.
elliptical boundaries, etc. Finite difference expressions at, and adjacent to, the boundary may then be applied using only grid points on the intersections of coordinate lines, without the need for any interpolation between points of the grid.

The avoidance of interpolation is particularly important for boundaries with strong curvature or slope discontinuities, both of which are common in physical applications. Likewise, interpolation between grid points not coincident with the boundaries is particularly inaccurate with differential systems that produce large gradients in the vicinity of the boundaries, and the character of the solution may be significantly altered in such cases. In most partial differential systems the boundary conditions are the dominant influence on the character of the solution, and the use of grid points not coincident with the boundaries thus places the most inaccurate difference representation in precisely the region of greatest sensitivity. The generation of a curvilinear coordinate system with coordinate lines coincident with all boundaries (herein called a "boundary-fitted coordinate system" for purposes of identification) is thus an essential part of a general numerical solution of a partial differential system.

A general method of generating boundary-fitted coordinate systems is to let the curvilinear coordinates be solutions of an elliptic partial differential system in the physical plane, with Dirichlet boundary conditions on all boundaries. One coordinate is specified to be constant on each of the boundaries, and a monotonic variation of the other coordinate around each boundary is specified. Thus, there is a coordinate line coincident with each boundary. The procedure is not restricted to two dimensions, allows the coordinate lines
to be concentrated as desired, and is applicable to all multi-connected regions (and thus to fields containing any number of arbitrarily shaped bodies).

The coordinate system so generated is not necessarily orthogonal, but orthogonality is not required, and its lack only requires that the partial differential system to be solved on the coordinate system when generated must be transformed directly through implicit partial differentiation rather than by use of the scale factors and differential operators developed for orthogonal curvilinear systems. An orthogonal system cannot be achieved with arbitrary spacing of the coordinate lines, and the capability for such concentration of coordinate lines is of more importance than orthogonality.

This general idea has been applied previously to two-dimensional regions interior to a closed boundary (simply-connected regions) by Winslow [1], Barfield [2], Chu [3], Amsden and Hirt [4], and Godunov and Prokopov [5]. Winslow [1] and Chu [3] took the transformed coordinates to be solutions of Laplace's equation in the physical plane which, as is shown in the next section, makes the physical cartesian coordinates solutions of a quasi-linear elliptic system in the transformed plane. Barfield [2] and Amsden and Hirt [4] reversed the procedure, taking the physical coordinates to be solutions in the transformed plane of a linear elliptic system which consists of Laplace's equation modified by a multiplicative constant on one term. This makes the transformed coordinates solutions of a quasi-linear elliptic system in the physical plane. Barfield also considered a hyperbolic system, but such a system cannot be used to treat general closed boundaries, since only elliptic systems allow specification of boundary conditions on the entirety of closed boundaries. Stadius [6] also used a hyperbolic system to generate a coordinate system for a doubly-connected region having parallel inner and outer boundaries.
With parallel boundaries it is only necessary to specify conditions on one of the boundaries, the location of the other boundary being free. The elliptic system, however, allows all boundaries to be specified as desired and thus has much greater flexibility.

Amsden and Hirt [4] constructed the coordinate generation method by iterative weighted averaging of the values of the physical coordinates at fixed points in the transformed plane in terms of values at neighboring points. Although not stated as such, this procedure is precisely equivalent to solving Laplace's equation, or modification thereof of the form noted above in Barfield [2], for the physical coordinates in the transformed plane by Gauss-Seidel iteration. Amsden and Hirt also allowed the boundary to move at each iteration, but this is simply equivalent to approaching the solution of the boundary-value problem through a succession of boundary-value problems converging to the problem of interest. In the approach of Godunov and Prokopov [5] the elliptic system is quasi-linear in both the physical and transformed planes. These authors applied a second transformation to that used by Chu [3], the transformation functions of this latter transformation being chosen a priori to control the coordinate spacing. Though not stated as such, the overall transformation may be shown to be generated by taking the transformed coordinates to be solutions in the physical plane of Laplace's equation modified by the addition of a multiple of the square of the Jacobian, the multiplicative factors being a priori chosen functions of the physical coordinates.

Meyder [7] generated an orthogonal curvilinear system by solving for the potential and "force" lines in a simply-connected region and taking these as the coordinate lines. This amounts to making the curvilinear coordinates
solutions of Laplace equations in the physical plane with Dirichlet boundary conditions (constant) on part of the boundary and Neumann boundary conditions (vanishing normal derivative) on the remainder. The solution for the coordinates was done, however, in the physical plane on a rectangular grid using interpolation at the curved boundaries, rather than in the transformed plane.

Orthogonal curvilinear coordinates for multi-connected regions, including regions with two bodies, have been generated by Ives [8] using conformal mapping. Conformal mapping is a special case of the generation of coordinate systems by solving an elliptic boundary value problem, but is not extendable to three dimensions and is less flexible in the spacing of the coordinate lines.

There have also been a number of transformations developed directly for special purposes without solving a partial differential system. One such approach is that of Gal-Chen and Somerville [9] for the treatment of an irregular boundary, such as mountaneous terrain, in a simply-connected region.

In the present research, the technique of generating the transformed coordinates as solutions of an elliptic differential system in the physical plane has been applied to multi-connected regions with any number of arbitrarily shaped bodies (or holes). The elliptic equations for the coordinates are solved in finite difference approximation by SOR iteration. Procedures for controlling the coordinate system so that coordinate lines can be concentrated as desired have been developed. Present effort is confined to two dimensions in the interest of computer economy, but the technique is extendable in principle to three dimensions. The procedure is also applicable
to fields with time-dependent boundaries, one coordinate line remaining fixed to the moving boundary. Here the equations for the coordinates must be re-solved at each time step. The computational grid remains fixed in spite of the movement of the physical grid.

Any partial differential system can be solved on the boundary-fitted coordinate system by transforming the set of partial differential equations of interest, and associated boundary conditions, to the curvilinear system. (It is shown in Appendix B that the equations do not change type, i.e., elliptic, parabolic, hyperbolic, under the transformation.) Since the boundary-fitted coordinate system has coordinate lines coincident with the surface contours of all bodies present, all boundary conditions can be expressed at grid points, and normal derivatives on the bodies can be represented using only finite differences between grid points on coordinate lines, without need of any interpolation, even though the coordinate system is not orthogonal at the boundary. The transformed equations can then be approximated using finite difference expressions and solved numerically in the transformed plane. Thus, regardless of the shape of the physical boundaries, and regardless of the spacing of the finite grid in the physical field, all computations, both to generate the coordinate system and, subsequently, to solve the partial differential system of interest can be done on a rectangular field with a square mesh with no interpolation required on the boundaries. Moreover, the physical boundaries may even be time-dependent without affecting the grid in the transformed region.

The computer software utilized to generate the boundary-fitted coordinate system is independent of the set of partial differential equations to be solved on this system. For example, numerical solutions for inviscid
and viscous fluid flows have been obtained using this system (Ref. 10-14).
The partial differential equations governing these phenomena differ drastically. However, for a given body geometry, the same boundary-fitted system generation program was used in both solutions. Another major advantage of using boundary-fitted coordinates is that the computer software generated to approximate the solution of a given set of partial differential equations is completely independent of the physical geometry of the problem. The coordinate systems for the wide variety of bodies included in this report, for example, were all developed utilizing the same computer program. Finally, it is shown in Appendix C that physical integral conservation relations need not be lost in the transformed plane.

This report presents a detailed development of the method for generation of boundary-fitted coordinate systems for general, multi-connected, two-dimensional regions. The basic doubly-connected region transformation is discussed in Section II in some detail. Extensions of the basic transformation to multi-connected regions, contracted coordinate systems, and time-dependent systems are also discussed. The numerical techniques used to implement the method and a number of specific examples are presented in the Sections III and IV. Examples of application to the solution of partial differential equations are given in Section V. Finally, instructions for use and a listing of the FORTRAN program are given in Section VI. Various derivative relations used in the transformation of partial differential systems and program parameters used in the examples included are given in Appendix A.

II. MATHEMATICAL DEVELOPMENT

Preliminaries

The general transformation from the physical plane \([x,y]\) to the transformed plane \([\xi,\eta]\) is given by the vector-valued function
The Jacobian matrix for this transformation is

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi(x,y) \\ \eta(x,y) \end{bmatrix}$$

(1)

where the subscripts denote partial differentiation in the usual manner.

The inverse function or transformation of (1) is, if it exists,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix}$$

(3)

The Jacobian matrix of (3) will be denoted by $J_2$ and is given by

$$J_2 = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}$$

(4)

The Jacobian determinant, or Jacobian as it is normally called, is then

$$J = \det[J_2] = x_\xi y_\eta - x_\eta y_\xi$$

(5)

The Jacobian matrices, (2) and (4), are related by

$$J_1 = [J_2]^{-1}$$

(6)

which implies the relations

$$\xi_x = y_\eta / J, \quad \xi_y = -x_\eta / J, \quad \eta_x = -y_\xi / J, \quad \eta_y = x_\xi / J$$

(7a,b,c,d)
Partial derivatives are transformed as follows:

\[
\frac{f}{x} = \frac{\partial (f,x)}{\partial (\xi,\eta)} / \frac{\partial (x,y)}{\partial (\xi,\eta)} = \frac{\frac{y f_x - y f_s}{\eta} \xi}{J}
\]  

(8)

\[
\frac{f}{y} = \frac{\partial (x,f)}{\partial (\xi,\eta)} / \frac{\partial (x,y)}{\partial (\xi,\eta)} = \frac{\frac{-x f_x + x f_s}{\eta} \xi}{J}
\]  

(9)

where \( f \) is some sufficiently differentiable function of \( x \) and \( y \). Higher derivatives are obtained by repeated application of (8) and (9). A comprehensive set of transformed derivatives, operators, unit vectors, and other useful relations is given in Appendix A.

Sufficient conditions for the transformation described above to exist are given by the inverse function theorem (Ref. 15). In particular, if the component functions of (1) are continuously differentiable at a point, say \( (x_o, y_o) \), and the Jacobian matrix (2) is nonsingular at \( (x_o, y_o) \) then there exists a disk \( N_o \) about \( (x_o, y_o) \) such that the inverse function (3) exists and (6) holds for all \( [x,y] \) in \( N_o \). It is readily apparent that the theorem guarantees existence only in a local fashion. For this reason component functions of (1) which possess even more desirable properties than those required by the inverse function theorem are sought.

Since the basic idea of the present transformation is to let the component functions of (1) be solutions of an elliptic Dirichlet boundary value problem, an obvious choice is to require that \( \xi(x,y) \) and \( \eta(x,y) \) be either harmonic, subharmonic, or superharmonic. Harmonic functions have continuous derivatives of all orders. Moreover, harmonic functions obey a maximum principle, which states that the maximum and minimum values of the function must occur on the boundaries of the region \( D \). Thus, since no extrema occur within \( D \), the first derivatives of the function will not simultaneously
vanishes in \( D \), and hence the Jacobian \( J \) will not be zero due to the presence of an extremum. (Note that this merely removes one condition which may cause the Jacobian to vanish.) Further, the maximum principle guarantees uniqueness of the coordinate functions \( \xi(x,y) \) and \( \eta(x,y) \), (Ref. 16), and thus ensures that no overlapping of the boundaries will occur. Subharmonic and superharmonic functions are also continuously differentiable and obey a maximum principle. (The maximum principle is not as strong for these functions as it is for harmonic functions.) A more general discussion of the mathematical properties of the transformation is given in [17].

Doubly-Connected Region

Consider the transformation of a two-dimensional, doubly-connected region \( D \) bounded by two simple, closed, arbitrary contours onto a rectangular region \( D^* \) as shown in Figure 1. (The basic transformation is discussed here assuming that the body contour and outer boundary are transformed, respectively, to the constant \( \eta \)-lines forming the bottom and top sides of the transformed region. The more general case of segmented body contours transforming to any side of the transformed region follows analogously and is discussed in later sections. The computer program allows the body contour(s) and outer boundary to be segmented and placed around the sides of the transformed plane in any manner desired.) Let \( \Gamma_1 \) map onto \( \Gamma_1^* \), \( \Gamma_2 \) map onto \( \Gamma_2^* \), \( \Gamma_3 \) onto \( \Gamma_3^* \), and \( \Gamma_4 \) onto \( \Gamma_4^* \). For identification purposes region \( D \) will be referred to as the physical plane, \( D^* \) as the transformed plane, and \( \Gamma_1 \) as the body contour. Note that the transformed boundaries \( (\Gamma_1^* \text{ and } \Gamma_2^*) \) are made constant coordinate lines \( (\eta \text{-lines}) \) in the transformed plane. The contours \( \Gamma_3 \) and \( \Gamma_4 \) which connect the contours \( \Gamma_1 \) and \( \Gamma_2 \) are coincident in the physical
plane and thus constitute re-entrant boundaries in the transformed plane.

In view of the closing remarks of the previous section, consider taking Laplace's equation as the generating elliptic system. That is, let $\xi(x,y)$ and $\eta(x,y)$ be harmonic in $D$. Then

$$\xi_{xx} + \xi_{yy} = 0 \quad (10a)$$

$$\eta_{xx} + \eta_{yy} = 0 \quad (10b)$$

with the Dirichlet boundary conditions

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi_1(x,y) \\ n_1 \end{bmatrix}, \ [x,y] \in \Gamma_1 \quad (10c)$$

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi_2(x,y) \\ n_2 \end{bmatrix}, \ [x,y] \in \Gamma_2 \quad (10d)$$

where $n_1$ and $n_2$ are different constants ($n_2 > n_1$), and $\xi_1(x,y)$ and $\xi_2(x,y)$ are specified monotonic functions on $\Gamma_1$ and $\Gamma_2$, respectively, varying over the same range. The arbitrary curve joining $\Gamma_1$ and $\Gamma_2$ in the physical plane, which transforms to the right and left sides of the transformed plane, specifies a branch cut for the multiple-valued function $\xi(x,y)$. Thus, the values of the physical coordinate functions $x(\xi,\eta)$ and $y(\xi,\eta)$ are the same on $\Gamma_3$ as on $\Gamma_4$, and these functions and their derivatives are continuous from $\Gamma_3$ to $\Gamma_4$. Therefore, boundary conditions are neither required nor allowed on $\Gamma_3$ and $\Gamma_4$. (A graphic analog to the above ideas can be found in most complex variable texts where Riemann surfaces are discussed. For example, see Levinson and Redheffer, Ref. 16.)
Since it is desired to perform all numerical computations in the uniform rectangular transformed plane, the dependent and independent variables must be interchanged in (10). Use of equation (A.18) of Appendix A yields the coupled system

\begin{align*}
ax_\xi - 2bx_\xi_\eta + yx_\eta_\eta &= 0 \\
ay_\xi - 2by_\xi_\eta + yy_\eta_\eta &= 0
\end{align*}

where

\begin{align*}
\alpha &\equiv x_\eta^2 + y_\eta^2 \\
\beta &\equiv x_\xi x_\eta + y_\xi y_\eta \\
\gamma &\equiv x_\xi^2 + y_\xi^2
\end{align*}

with the transformed boundary conditions

\begin{align*}
\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} f_1(\xi, \eta_1) \\ f_2(\xi, \eta_1) \end{bmatrix}, \ [\xi, \eta_1] \in \Gamma_1^* \\
\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} g_1(\xi, \eta_2) \\ g_2(\xi, \eta_2) \end{bmatrix}, \ [\xi, \eta_2] \in \Gamma_2^*
\end{align*}

The functions \( f_1(\xi, \eta_1) \), \( f_2(\xi, \eta_1) \), \( g_1(\xi, \eta_2) \), and \( g_2(\xi, \eta_2) \) are specified by the known shape of the contours \( \Gamma_1 \) and \( \Gamma_2 \) and the specified distribution of \( \xi \) thereon. As noted, boundary data are neither required nor allowed along the re-entrant boundaries \( \Gamma_3^* \) and \( \Gamma_4^* \).

The system given by the equations of (11) is a quasi-linear elliptic system for the physical coordinate functions, \( x(\xi, \eta) \) and \( y(\xi, \eta) \), in the
transformed plane. Although this system is considerably more complex than that given by (10), the boundary conditions \((llf,g)\) are specified on straight boundaries, and the coordinate spacing in the transformed plane is uniform. The boundary-fitted coordinate system generated by the solution to (11) has a constant \(\eta\)-line coincident with each boundary in the physical plane. The \(\xi\)-constant lines may be spaced as desired around the boundaries since the assignment of the \(\xi\)-values to the \([x,y]\) boundary points via the functions \(f_1', f_2', g_1',\) and \(g_2\) in \((llf,g)\) is arbitrary. (Numerically the discrete boundary values \([x_k,y_k]\) are transformed to equi-spaced discrete \(\xi_k\)-points on both boundaries.) Control of the radial spacing of the \(\eta\)-constant lines and of the incidence angle of the \(\xi\)-constant lines at the boundaries is accomplished by varying the generating elliptic system (i.e., the system of which \(\xi(x,y)\) and \(\eta(x,y)\) are solutions) as will be demonstrated in Section IV. As illustrated in Figure 1, the left and right boundaries of the transformed plane are re-entrant boundaries, which implies that both solutions, \(x(\xi,\eta)\) and \(y(\xi,\eta)\), are required to be periodic in the region \([[\xi,\eta]|-\infty < \xi < \infty, \eta_1 \leq \eta \leq \eta_2}\].

Multiply-Connected Region

The basic ideas and procedures introduced in the preceding section can be extended to regions containing more than one body—that is, to general multi-connected or multi-body regions. One transformation for two bodies is illustrated in Figure 2. The bodies are connected with one arbitrary cut, with an additional cut joining one of the body contours to the outer boundary. The physical plane contours \(\Gamma_1 - \Gamma_8\) map respectively onto the contours \(\Gamma_1^* - \Gamma_8^*\) in the transformed plane. Note that the body defined by the union of \(\Gamma_7\) and \(\Gamma_8\) is split into two segments \((\Gamma_7^* and \Gamma_8^*)\), as is the cut joining
this body and the one defined by contour $\Gamma_1$. The $n$-coordinate is the same for both of the bodies and cut between them. Conversely, the cut defined by $\Gamma_3$ and $\Gamma_4$ in the physical plane is taken as a $\xi$-constant line in the transformed plane as before for the single body case. The outer boundary contour $\Gamma_2$ maps onto the upper boundary in the $[\xi, n]$ plane, becoming a constant $n$-line in the manner of the one-body transformation. In contrast to the one-body transformation, two re-entrant boundaries occur for this two-body transformation. The left and right vertical boundaries ($\Gamma_4^*$, $\Gamma_3^*$) appear as before. In addition a horizontal re-entrant segment due to the coincidence of $\Gamma_5$ and $\Gamma_6$ in the physical plane arises. The coordinate functions and the derivatives thereof are thus continuous across these re-entrant boundaries.

The boundary-fitted coordinates for the multi-body transformations are again determined by the solution of the set of equations (11) with the added boundary conditions

\[
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
= \begin{bmatrix}
  h_1(\xi, n_1) \\
  h_2(\xi, n_1)
\end{bmatrix}, \quad [\xi, n_1] \in \Gamma_7^*
\]

(11h)

\[
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
= \begin{bmatrix}
  q_1(\xi, n_1) \\
  q_2(\xi, n_1)
\end{bmatrix}, \quad [\xi, n_1] \in \Gamma_8^*
\]

(11i)

to define the additional body. Note that boundary conditions cannot be specified along the re-entrant boundaries defined by $\Gamma_3^*$, $\Gamma_4^*$, $\Gamma_5^*$, and $\Gamma_6^*$. As with the basic transformation all numerical computations both to generate the system and subsequently to utilize the coordinates for
solving a set of partial differential equations, are executed on a rectangular field with a uniform grid.

Simply-Connected Region

For a simply-connected region there are no bodies in the field and hence, no cuts in the physical plane and no re-entrant boundaries in the transformed plane. The single continuous boundary surrounding the physical field transforms to the entire rectangular boundary of the transformed field. The manner in which the physical boundary is split into four segments for placement on the four sides of the rectangular boundary in the transformed plane is a matter of choice.

Coordinate System Control

Control of the spacing of the coordinate lines on the body is easily accomplished since the points on the body are input to the program. The spacing of the coordinate lines in the field, however, must be controlled by varying the elliptic generating system for the coordinates. One method of variation is to modify the Laplace equations (10) by adding inhomogeneous terms to the right sides so that the generating system becomes

$$
\xi_{xx} + \xi_{yy} = P(\xi, \eta), \quad \eta_{xx} + \eta_{yy} = Q(\xi, \eta)
$$

(12)

In the transformed plane these equations become

$$
\alpha_x \xi_{\xi} - 2\beta_x \xi_{\eta} + \gamma_x \eta_{\eta} + J^2(P_x \xi + Q_x \eta) = 0
$$

(13a)

$$
\alpha_y \xi_{\xi} - 2\beta_y \xi_{\eta} + \gamma_y \eta_{\eta} + J^2(P_y \xi + Q_y \eta) = 0
$$

(13b)
The effect of changing the functions $P(\xi, \eta)$ and $Q(\xi, \eta)$ on the coordinate system can be seen by examining the system (12). If $P = Q = 0$, the system reduces to the pair of Laplace equations used as the original generating system, i.e., Eq. (10). Let $\xi', \eta'$ be the solutions of (10) with boundary values on $D$. Let $\xi'', \eta''$ be the solutions to system (12) with the same boundary values and with positive functions $P$ and $Q$ on the right hand side of the equations. Now $\xi''$ and $\eta''$ are subharmonic on $D$. Thus for any constant $k$, the $\xi'' = k$ coordinate line would be closer to $\xi'' = \xi_{\text{max}}$ than would the $\xi' = k$ line. The same relation holds for the $\eta'' = k$ and $\eta' = k$ coordinate lines.

Now if all or part of the curve $\xi' = \xi_{\text{max}}$ is a branch cut in $D$, this branch cut will move in the direction of increasing $\xi$ to meet the curve $\xi'' = \xi_{\text{max}}$ if the right side of the system (12) is increased from 0 to a positive function $P$. An analogous discussion can be made on the effect of negative functions $P$ and $Q$ on the generated curvilinear coordinate system.

Even though a change in $P$ or $Q$ in a subregion of $D$ would change the coordinate lines throughout the entire region, the effect would certainly be more pronounced in the subregion. Also, the greater the change in $P$ and $Q$, the greater the movement of the coordinate lines. By varying the sign and magnitude of $P(\xi, \eta)$ and $Q(\xi, \eta)$ for different values of $\xi$ and $\eta$, considerable control can be exerted over the coordinate line spacing as will be seen from the examples that follow in Section IV.

One particularly effective procedure is to choose $P$ and $Q$ as exponential terms, so that the coordinates are generated as the solutions of
\[ \xi_{xx} + \xi_{yy} = -\sum_{i=1}^{n} a_i \text{sgn}(\xi - \xi_i) \exp(-c_i |\xi - \xi_i|) \]

\[ - \sum_{j=1}^{m} b_j \text{sgn}(\xi - \xi_j) \exp(-d_j \sqrt{(\xi - \xi_j)^2 + (n - n_j)^2}) \]

\[ \equiv P(\xi, \eta) \quad (14a) \]

\[ \eta_{xx} + \eta_{yy} = -\sum_{i=1}^{n} a_i \text{sgn}(\eta - \eta_i) \exp(-c_i |\eta - \eta_i|) \]

\[ - \sum_{j=1}^{m} b_j \text{sgn}(\eta - \eta_j) \exp(-d_j \sqrt{(\xi - \xi_j)^2 + (n - n_j)^2}) \]

\[ \equiv Q(\xi, \eta) \quad (14b) \]

where the positive amplitudes and decay factors are not necessarily the same in the two equations. Here the first terms have the effect of attracting the \( \xi = \) constant lines to the \( \xi = \xi_i \) lines in Equation (14a), and attracting \( \eta = \) constant lines to the \( \eta = \eta_i \) lines in Equation (14b). The second terms cause \( \xi = \) constant lines to be attracted to the points \( (\xi_j, \eta_j) \) in (14a), with similar effect on \( \eta = \) constant lines in (14b). No computational difficulties have been encountered because of the discontinuities in \( P \) and \( Q \) caused by the \text{sgn} function, which is defined by setting \text{sgn}(x) to be 1, 0, or -1 depending on whether \( x \) is positive, zero, or negative. Should problems arise in later applications, this function can be replaced by \( \frac{2}{\pi} \text{Arctan}(nx) \), where \( n \) is a large positive integer. The \text{sgn} function was chosen to give the maximum control. Several examples of the use of coordinate system control are given in Section IV.
With the inclusion of the sgn function (or the arctan function) the equations (14a & b) for the curvilinear coordinates are no longer subharmonic or superharmonic, since the sgn function causes a sign change on the right side when the attraction is to lines or points not on the boundaries. It is possible, therefore, that too strong an attraction amplitude may cause the system to overlap and therefore be unusable. Many successful systems (all those included in the examples given herein) have been generated, however, using the above equations.

The use of the sign-changing sgn function is only necessary to cause attraction to both sides of a line or point in the field. Elimination of this function causes attraction on one side and repulsion on the other. If it is only desired to concentrate coordinate lines near one boundary, such as the body surface, then there is no need for the sign change, and the sgn function can be eliminated. In this case the equations are subharmonic or superharmonic, and a maximum principle is in effect to prevent overlap. Such a choice is provided for in the computer program.

The subject of coordinate system control is still very much under investigation, and other control functions are being evaluated. It is anticipated that new coordinate control packages will be made available for inclusion in the code when warranted. Of particular interest is the capability to cause a specified number of lines to fall within a certain physical region, such as a boundary layer. Another area of further investigation is the coupling of the coordinate equations with the partial differential system to be solved thereon so that the coordinate lines concentrate automatically in regions of high gradient.
Time-Dependent Coordinate Systems

If the coordinate system changes with time then the grid points move in the physical plane. Ordinarily such movement of the physical grid points would require interpolation among the grid points to produce values of the dependent variables at the new locations of the grid points. With the present method of coordinate system generation, however, it is possible to perform all computation on the fixed rectangular grid in the transformed plane without any interpolation no matter how the grid points move in the physical plane as time progresses. This occurs as follows:

Recall that the coordinate system is generated as the solution of some elliptic system with the values of the transformed coordinates \([\zeta, \eta]\) specified on the boundaries in the physical plane, one of these coordinates being specified to be constant on the boundaries and the other being distributed as desired along the boundaries in order, perhaps, to concentrate grid points in certain regions. The transformed coordinates define a rectangular plane, the extent of which is determined by the range of the values of \(\zeta\) and \(\eta\). Now if the same boundary values of \(\zeta\) and \(\eta\) are redistributed in the physical plane, perhaps because the boundaries in the physical plane have actually moved or maybe just to change the concentration of grid points around the boundaries, and the elliptic system is re-solved for the transformed coordinates with these new boundary conditions, new transformation functions can be produced with still the same range of values in \(\zeta\) and \(\eta\) (provided the elliptic system used exhibits a maximum principle) and hence to the same rectangular field in the transformed plane. The grid points in the rectangular transformed plane thus remain stationary, and the effect of the movement of the coordinate system in the physical plane is just to change the values of the physical coordinates \([x, y]\) at the fixed grid points in the rectangular transformed plane.
Thus, although the position of a grid point changes on the physical plane, its position in the transformed plane is fixed. The time derivative transforms to the transformed plane as shown below:

\[
\frac{\partial f}{\partial t} = \frac{\partial (x,y,f)}{\partial (\xi,\eta,t)} / \frac{\partial (x,y,t)}{\partial (\xi,\eta,t)}
\]

\[
= f_t - x \left( f_\xi \eta - f_\eta \xi \right) / J
\]

\[
+ y \left( f_\xi \eta - f_\eta \xi \right) / J
\]

(15)

Here all derivatives are expressed in the transformed plane, so that the interpolation that would be necessary to supply values at grid points in the physical plane that have moved is not required in the transformed plane. (Note that in the transformed expression for the time derivative, all derivatives are taken at the fixed grid points in the transformed plane. The movement of the grid in the physical plane is reflected only through the rates of change of \( x \) and \( y \) at the fixed grid points in the transformed plane.)

The problem of solving \( N \) partial differential equations of any type in a physical region with time dependent boundaries has been replaced by a new problem consisting of \( N + 2 \) equations with fixed boundaries. The two additional equations are, of course, those governing the transformation (either (10) or (12)). Thus, it is possible to construct numerical solutions to physical problems with time dependent boundaries in a fixed rectangular plane with a fixed square mesh with no interpolation required. Again the above statements imply the independence of computer software from physical geometry. Problems involving moving blast fronts, shocks, free surfaces, and any other time-varying boundaries can be attacked successfully with these
procedures. (Some application to free surface flow is illustrated in Ref. 26.) In addition this method allows time-dependent concentration of grid points as desired in the physical plane.

The use of time-dependent coordinate systems requires that the difference equations for the curvilinear coordinates be resolved at each time step, of course. The coordinate values at the previous time step can serve as the initial guess for the next, however, so that the iteration will converge rapidly.

III. NUMERICAL SOLUTION

Difference Equations

The discussion here assumes the body contour and outer boundary transform, respectively, to the $\eta$-lines forming the lower and upper sides of the rectangular transformed plane as discussed in Section II. More general segmentation of the body contour(s) and the outer boundary, and placement as desired around the boundary of the transformed field, are provided for in the computer program.

The finite difference grid for the single-body problem is illustrated in Figure 3a. Circles denote the points at which the difference approximation to (11a,b) or (13a,b), are applied, while the triangles denote boundary points. The left and right vertical boundaries are coincident in the physical plane, and the values of $x$ and $y$ are thus equal along these lines. Such lines are designated re-entrant boundaries as indicated before. If the number of $\xi = \text{constant}$ lines is designated IMAX and the number of $\eta$-lines by JMAX, the computational field size is $(\text{JMAX}-2)(\text{IMAX}-1)$. Boundary values are specified
on $j=1$ and $j=\text{JMAX}$ for all $1 \leq i \leq \text{IMAX}$. The $j=1$ line corresponds to
the contour $\Gamma_1$ (the body contour) in the physical plane while $j=\text{JMAX}$ is
associated with the remote boundary contour $\Gamma_2$. Second-order central dif-
fference expressions (see Appendix D) are used to approximate all derivatives
in the transformed equations. The resulting equations are, for (11a,b),

$$
x_{i,j} = \left[ a'_{i,j} (x_{i-1,j} + x_{i+1,j}) - \frac{\beta'_{i,j} (x_{\xi,1})_{i,j}}{2} \right] + \gamma'_{i,j} (x_{i,j+1} + x_{i,j-1}) \right] / \left[ 2(a'_{i,j} + \gamma'_{i,j}) \right] (16a)
$$

$$
y_{i,j} = \left[ a'_{i,j} (y_{i-1,j} + y_{i+1,j}) - \frac{\beta'_{i,j} (y_{\xi,1})_{i,j}}{2} \right] + \gamma'_{i,j} (y_{i,j+1} + y_{i,j-1}) \right] / \left[ 2(a'_{i,j} + \gamma'_{i,j}) \right] (16b)
$$

where $a'_{i,j}$, $\beta'_{i,j}$, $\gamma'_{i,j}$, $(x_{\xi,1})_{i,j}$, and $(y_{\xi,1})_{i,j}$ are the difference ap-
proximations for $\alpha$, $\beta$, $\gamma$, and the cross derivatives respectively. These
expressions are developed in Appendix D. The re-entrant boundaries
occurring at $i=1$ and $i=\text{IMAX}$ are dealt with as follows. Since the values
of $x$ and $y$ are equal along these lines, iteration is necessary along only
one of them. Choosing $i=1$ for convenience, the $\xi$-derivatives along this
line are approximated as exhibited below:

$$
(x_{\xi})_{i,j} = \left( x_{2,j} - x_{\text{IMAX}-1,j} \right) / 2 \tag{17a}
$$

$$
(x_{\xi x})_{i,j} = x_{2,j} - 2x_{1,j} + x_{\text{IMAX}-1,j} \tag{17b}
$$

$$
(x_{\xi n})_{i,j} = \left( x_{2,j+1} - x_{2,j-1} + x_{\text{IMAX}-1,j-1} - x_{\text{IMAX}-1,j+1} \right) / 4 \tag{17c}
$$
for \( 2 \leq j \leq \text{JMAX}-1 \). Similar expressions are used for the derivatives of \( y \). The set of non-linear simultaneous difference equations produced by (16a,b) is solved by point SOR iteration.

**Multiple-Body Fields**

A sample mesh for a two-body transformation is shown in Figure 3b. As before, circles denote computational nodes and triangles the boundary points. Values defining the outer boundary contour \( \Gamma_2 \) (see Figure 2) are required for \( 1 \leq i \leq \text{IMAX} \) along the \( j=\text{JMAX} \) line. Boundary values specifying the shape of the body contours \( \Gamma_1, \Gamma_7, \) and \( \Gamma_8 \) in the physical plane are required in segments along the \( j=1 \) line as follows:

\[
\begin{align*}
\Gamma_1: & \quad I2 \leq i \leq I3 \\
\Gamma_7: & \quad 1 \leq i \leq I1 \\
\Gamma_8: & \quad I4 \leq i \leq \text{IMAX}
\end{align*}
\]

The above requirements may be more clearly seen by comparing Figures 2 and 3b. The difference equations (16a,b) and the vertical re-entrant boundary relations (17a,b,c) are also valid for multi-body transformations. The primary difference in the two transformations results from the existence, in the multi-body case, of the horizontal re-entrant boundaries along \( j=1 \). The equivalence of \( x \) and \( y \) values along these segments is demonstrated in Figure 3b. Again it is only required to calculate values along one re-entrant segment. Choosing the leftmost \( (I1 \leq i \leq I2) \), the \( n \)-derivatives are calculated using special procedures which are illustrated by the expressions below for the point marked with an \( \oplus \) \((i,j = I1 + 1,1)\) in Figure 3b:
\[
(x_n)_{l+1,1} = (x_{l+1,2} - x_{l+1,1})/2 \quad (18a)
\]
\[
(x_{nn})_{l+1,1} = x_{l+1,2} - 2x_{l+1,1} + x_{l+1,2} \quad (18b)
\]
\[
(x_{nn})_{l+1,1} = (x_{l+2,2} - x_{l+1,2}^2 + x_{l+1,2} - x_{l+1,2})/4 \quad (18c)
\]

Note the existence of the horizontal re-entrant boundaries increases the size of the computational field somewhat. In the two-body example given the number of additional equations to be solved is \(12 - (l+1)\).

Multiple-Body Segment Arrangements

In the case of a single body it is logical to keep the body contour in one segment, with a single cut connecting the single segment to the outer boundary. This type of arrangement is illustrated in Figure 4a. (Figure 4b shows an alternate single body arrangement. In these and all subsequent figures the dotted lines on the segment arrangement diagrams identify the two members of a re-entrant pair.) In the case of multiple bodies there is a wider choice of reasonable arrangements, some of which may be better than others for certain applications. The boundaries in the physical plane may be split into as many segments as desired, and these segments may be arranged around the rectangular boundary of the transformed plane in any way desired. These segments are all connected by branch cuts in the physical plane and by re-entrant boundaries in the transformed plane. Several of these arrangements are illustrated in Figures 5-13. Illustrative values of the segment input parameters are given in Table 1 for each of these arrangements, and input instructions are given in Section VI.
In the arrangement of Figure 5, an \( \eta \)-line encircles both bodies and forms a cut between the bodies, the cut to the outer boundary being a \( \xi \)-line. The outer boundary is also a line of constant \( \eta \) but at a different value. Here one body is split into two segments, while the other body and the outer boundary are each in single segments. Figure 6 shows an arrangement in which each body is in a single segment, each body being a \( \xi \)-line of different value. Here there is no cut between the bodies, but rather an \( \eta \)-line cut between each body and the outer boundary, which is split into two segments, each being an \( \eta \)-line of different value. (This produces a system similar to a bi-polar coordinate system.) In Figure 7 each body is also in a single segment with the outer boundary split into two segments, but here an \( \eta \)-line encircles each body and forms the cut between that body and the outer boundary. In Figure 8 one body is a single segment encircled by an \( \eta \)-line which forms a cut to the other body. The other body is split into two segments, each being a \( \xi \)-line of different value, with each segment connected to the outer boundary by a \( \xi \)-line. The outer boundary is in a single segment and is an \( \eta \)-line. Other arrangements are shown in Figures 9-13. All these arrangements are shown without coordinate line attraction, and, consequently, many of the resulting systems exhibit wide spacing in concave areas. This spacing can be improved by coordinate attraction as illustrated in the examples of Section IV. The results of the use of several of these multiple-body segment arrangements are given for two-body potential flow in Section V. Coordinate system control was used effectively in that case to improve the spacing in the concave region. These concave regions occur when the cut and body are on the same coordinate line.
Initial Guess

Since the difference equations are nonlinear, the initial guess must be within a certain neighborhood of the solution if the iterative solution is to converge. With some segment arrangements a logical choice of an initial guess is difficult to perceive. Therefore several different types of initial guesses have been inserted in the program, with the choice to be made by the user as guided by past experience. The rationale for some of these guesses is more intuitive than analytical. The choices available are detailed below. (In each case the initial values of x and y on all cuts are interpolated linearly between the boundary values at the cut end points.) The choice is controlled by the input parameter IGES as discussed in Section VI. The guess type is identified by this number in the discussion below.

(a) Weighted Average of Four Boundary Points - Here the values of x and y at each point in the field are set equal to the average of the four boundary points having either the same $\xi$ index or the same $\eta$ index, the average being weighted by the distance to the boundary in the transformed plane. Thus

$$2 x_{ij} = \frac{J_{\text{MAX}} - 1}{J_{\text{MAX}} - 1} x_{1,1} + \frac{j - 1}{J_{\text{MAX}} - 1} x_{i, J_{\text{MAX}}} + \frac{i}{I_{\text{MAX}} - 1} x_{1,j} + \frac{i - 1}{I_{\text{MAX}} - 1} x_{I_{\text{MAX}}, j}$$

for $i = 2, 3, \ldots, I_{\text{MAX}} - 1$ and $j = 2, 3, \ldots, J_{\text{MAX}} - 1$. An analogous equation is used for y. (IGES = 1). A variation of this type (and also of types (b), (c), & (e)) is provided by IGED, whereby the average may be restricted to only a two-point average in either the $\xi$ or $\eta$ direction. The direction chosen should be that which proceeds between the bodies and the outer boundary. This variation is useful with an outer boundary located on three sides.
(b) Same as (a), except zeroes replace the values on the cuts in the formation of the average. This type of initial guess is particularly effective with simply-connected regions, single-body fields, and multiple-body segment arrangements having the all body segments on one horizontal (vertical) side and the outer boundary a single segment on the other horizontal (vertical) side. (IGES = 0)

(c) Weighted Average of Body Segment Boundary Points Only - This type guess is the same as that of (a), except that boundary points on cuts are not included in the formation of the average, which therefore may be formed with fewer than four points. This type and the exponential projection below are widely effective. (IGES = 2)

(d) Moment Projection - Here the initial value at each field point is given by

\[ x_{ij} = \frac{\sum d_{ijk} \sum x_k - \sum d_{ijk} x_k}{N \sum d_{ijk}} \quad (20) \]

with \( d_{ijk} = \sqrt{(x_{ij} - x_k)^2 + (y_{ij} - y_k)^2} \)

Here \( x_k \) is the value at a point on the boundary of the transformed plane; \( d_{ijk} \) is the distance to that boundary point in the transformed plane; \( N \) is the total number of boundary points; and the summations extend over the entire boundary in the transformed plane (IGES = 3). A modification of this type omits the division by \( N \) (IGES = 4).

(e) Exponential Weighted Average of Body Segment Boundary Points - This guess is similar to that of (c) except that the weight in the average is exponential rather than linear. Increasing IGES causes the points to contract nearer the boundary segments corresponding
to the lowest values of $n$ and $f$. This type is most effective when strong coordinate attraction is used with a single-body field having the body located on the bottom or left side of the rectangular transformed field. (IGES > 4)

(f.) Exponential Projection - The initial value of $x$ is determined at each field point by

$$x_{ij} = \frac{\Sigma_{k}^{n} \exp(-|\text{IGES}| d_{ijk}/d_{o})}{\Sigma_{k}^{n} \exp(-|\text{IGES}| d_{ijk}/d_{o})}$$

(21)

where $d_{o}$ is the diagonal length of the transformed field, the other quantities having the same definitions as in (d) above.

(IGES < 0)

Examples of these initial guess types are shown in Figures 14-23 for the segment arrangements given in Figures 4-13. The value of IGES for each is given in the upper left corner of each plot. Table 2 lists the types for which convergence was obtained with each of these segment arrangements.

The most widely effective initial guess in these cases was the exponential projection. This type of guess produced convergence in the single-body case and in six of the nine two-body cases. The optimum decay factor for the exponential projection was 40 (IGES = -40) in most cases, with an optimum of 20 (IGES = -20) in a few cases.

Guess Type 2 (IGES = 2) also gave convergence in six of the nine two-body cases and in the single-body case. However, the number of iterations required was a bit larger than with the exponential projection. Type 2 gave convergence with one segment arrangement (Figure 9) for which the exponential projection gave divergence, but gave divergence for another arrangement (Figure 12) for which the exponential projection gave convergence.

Arrangements having the outer boundary in two segments tend to be the more difficult to converge. Of the four arrangements of this type (Figures,
6, 7, 12, 13), Guess Type 2 gave convergence for only one (Figure 13), while the exponential projection failed for two of the four. The two arrangements of Figures 6 and 7 proved to be particular recalcitrant, requiring a switch to a different size field in order to get convergence for the arrangement of Figure 6 and the introduction of a special guess for that of Figure 7.

The general suggestion for two-body cases is to use either exponential projection with a decay factor of about 40 or Guess Type 2. For simply-connected regions, single-body cases, and the two-body segment arrangements having both bodies on the same coordinate line, Guess Type 0 is generally more efficient, although the exponential weighted average or projection and Guess Type 2 will also give convergence. Arrangements having two bodies in single segments on opposite sides of the transformed plane are particularly difficult, and Guesses Type 1 and 4 may be used.

With some segment arrangements with multiple bodies, convergence can be achieved with a close outer boundary, but not with the outer boundary farther out. Therefore, provision has been made for initially converging the solution with a circular outer boundary close in and then constructing an initial guess for a field with a larger circular outer boundary from this solution by linear projection to the larger field. This process can be repeated as many times as desired with the outer boundary gradually being moved out to its desired position. Two types of movement are provided: (a) the outer boundary radius is doubled at each step, or (b) the outer boundary radius is increased linearly at each step. This provision should not be used unless necessary, and then the movement of (a) is to be preferred in general, with as few steps as will produce convergence.

Finally, with strong coordinate line attraction it may not be possible to achieve convergence from an initial guess that gives convergence without attraction. Provision therefore has been made whereby the attraction can be
added gradually, the converged solution for a small attraction becoming the initial guess for a case of stronger attraction. Two types of increase in attraction strength are provided: (a) the attraction amplitude is doubled at each step, or (b) the attraction amplitude is increased linearly at each step. This procedure is to be used only when necessary. In general, type (a) is preferred, with as few steps as will provide convergence. Further discussion of the use of the various initial guesses is given in the instructions of Section VI.

Convergence Acceleration

For a difference equation of the general form

\[ a_1(f_{i+1,j} + f_{i-1,j}) + a_2(f_{i,j+1} + f_{i,j-1}) + b_1(f_{i+1,j} - f_{i-1,j}) + b_2(f_{i,j+1} - f_{i,j-1}) + c f_{ij} + d_{ij} = 0 \]  

(\(i = 1, 2, \ldots, I; j = 1, 2, \ldots, J\))

with boundary values specified on \(i = j = 0\), \(i = I + 1\), and \(j = J + 1\), and \(a_1, a_2, b_1, b_2, c,\) and \(d\) constant, the optimum value of the SOR acceleration parameter \(\omega\) can be obtained in the case where \(a_1^2 < b_1^2\) and \(a_2^2 < b_2^2\), and in the case where \(a_1^2 > b_1^2\) and \(a_2^2 > b_2^2\), (Ref. 18). The optimum parameters in these two cases are as follows:

Case #1: \(a_1^2 > b_1^2\) and \(a_2^2 > b_2^2\)

\[ \omega = \frac{2}{1 + \sqrt{1 - \rho^2}} \]  

(over-relaxation, \(1 < \omega < 2\))  

(23)

Case #2: \(a_1^2 < b_1^2\) and \(a_2^2 < b_2^2\)

\[ \omega = \frac{2}{1 + \sqrt{1 + \rho^2}} \]  

(under-relaxation, \(0 < \omega < 1\))  

(24)
where

\[ \rho = 2 \sqrt{\left(\frac{a_1}{c}\right)^2 - \left(\frac{b_1}{c}\right)^2} \cos \frac{\pi}{1 + 1} \]

\[ + 2 \sqrt{\left(\frac{a_2}{c}\right)^2 - \left(\frac{b_2}{c}\right)^2} \cos \frac{\pi}{J + 1} \]  

(25)

In the remaining case where \( a_1^2 > b_1^2 \) and \( a_2^2 > b_2^2 \), no theoretical determination of the optimum acceleration parameter exists as yet.

Since the difference equations for the coordinate system are nonlinear, the above theory is not directly applicable. However, if the equations are considered as locally linearized then a local optimum acceleration parameter can be obtained which will vary over the field. It should be noted that the local linearization is applied only to the determination of the acceleration parameters, not to the actual solution of the difference equations.

Following this approach and neglecting the effect of the cross derivatives, the local constants in the above equations become

\[ a_1 = a \]
\[ a_2 = \gamma \]
\[ b_1 = \frac{J^2 p}{2} \]
\[ b_2 = \frac{J^2 Q}{2} \]
\[ c = -2(\alpha + \gamma) \]

so that locally optimum acceleration parameters are

Case #1: \( \alpha_{ij} > \frac{J_{ij}^2|p_{ij}|}{2} \) and \( \gamma_{ij} > \frac{J_{ij}^2|Q_{ij}|}{2} \)

\[ \omega_{ij} = \frac{2}{1 + \sqrt{1 - \rho_{ij}^2}} \]  

(over-relaxation)  

(26)
Case #2: \( \alpha_{ij} \leq \frac{J_{ij} |p_{ij}|}{2} \) and \( \gamma_{ij} \leq \frac{J_{ij} |Q_{ij}|}{2} \)

\[
\omega_{ij} = \frac{2}{1 + \sqrt{1 + \rho_{ij}^2}} \quad \text{(under-relaxation)}
\]  

(27)

where

\[
\rho_{ij} = \frac{1}{\alpha_{ij} + \gamma_{ij}} \left[ \sqrt{\alpha_{ij}^2 - \frac{J_{ij}^4 p_{ij}^2}{4}} \cos \left( \frac{\pi}{\text{IMAX} - 1} \right) + \sqrt{\gamma_{ij}^2 - \frac{J_{ij}^4 Q_{ij}^2}{4}} \cos \left( \frac{\pi}{\text{JMAX} - 1} \right) \right]
\]  

(28)

In the remaining case where \( \alpha \geq \frac{J^2 |p|}{2} \) and \( \gamma \leq \frac{J^2 |Q|}{2} \), not even a local optimum is available. The program allows a choice of strategy in this case: over-relaxation, under-relaxation, or a weighted average as follows:

First \( \rho_1 \) and \( \rho_2 \) are calculated from

\[
\rho_1 = \frac{1}{\alpha + \gamma} \sqrt{\alpha^2 - \frac{J^4 p^2}{4}} \cos \left( \frac{\pi}{\text{IMAX} - 1} \right)
\]  

(29a)

\[
\rho_2 = \frac{1}{\alpha + \gamma} \sqrt{\gamma^2 - \frac{J^4 Q^2}{4}} \cos \left( \frac{\pi}{\text{JMAX} - 1} \right)
\]  

(29b)

If over-relaxation is specified then \( \omega \) is calculated from Eq. (26) using \( \rho = \rho_1 + \rho_2 \). The same expression for \( \rho \) is used in Eq. (27) if under-relaxation is specified. If the weighted average is to be used then, using \( \rho_1 \) in place of \( \rho \), \( \omega_1 \) is calculated from Eq. (26) if \( \alpha \geq \frac{J^2 |p|}{2} \) or by Eq. (27) if \( \alpha < \frac{J^2 |p|}{2} \). Similarly, using \( \rho_2 \) in place of \( \rho \), \( \omega_2 \) is calculated from Eq. (26) if \( \gamma > \frac{J^2 |Q|}{2} \), else by Eq. (27). The average is then formed by

\[
\omega = \frac{\rho_1 \omega_1 + \rho_2 \omega_2}{\rho_1 + \rho_2}
\]  

(30)
Optimum Acceleration Parameters

In order to provide some guide to the selection of acceleration parameters for the most rapid convergence of the iterative solution, the optimum values were determined in a number of representative cases by computer experimentation.

The body for this study was a Karman-Trefftz airfoil, the contour of which is shown in Figure 24. The points on the contour were spaced at equal angular increments in the complex plane from which the airfoil was generated, with three additional points added at half, fourth, and eighth angular increments above and below the trailing edge to provide finer resolution in that region.

A basic case was selected and four quantities, (the number of points on the body, the number of coordinate lines surrounding the body, the amplitude of the coordinate line attraction to the body, and the radius of the circular outer boundary) were varied individually and in pairs above and below the basic values. The initial guess type (Type 0) giving the fastest convergence for the basic case was used for all. Each case was run to convergence of $10^{-4}$. Additional cases were run with different convergence criteria and different numbers of steps in the addition of the final attraction amplitude. The basic case was also run with a circular cylinder as the body for comparison of the effect of the body shape.

A series of two-body cases was also run with two of the same Karman-Trefftz airfoils positioned as an airfoil and flap system. Only one segment arrangement was considered, with both bodies on the same curvilinear coordinate line. The multiple-airfoil system and the segment arrangement are shown in Figure 25. The same set of quantities varied in the single body case was varied individually above and below the basic values (except
that variation of the number of points on the bodies above the basic value involved too much core and was omitted). The initial guess type was the same as that used for the single-body studies, which proved to give the fastest convergence in the basic two-body case as well. The values of all input parameters for the cases run are given in Tables 3-6.

The results of these studies are given in Tables 7-10. For the single-body field, Table 7 shows the effect of individual variation of each of the chosen quantities; Table 8 gives the effect of variation in pairs, and in Table 9 other miscellaneous quantities are varied. Table 10 gives the results for the double-body field. The number of iterations required with the variable acceleration parameter field and the average variable acceleration parameter over the field are also included in these tables. (Under-relaxation was used in the case of complex eigenvalues.) In a number of cases the variable acceleration parameters tend to be too large, and only in a few cases was the variable field better than the uniform experimentally determined optimum.

Plots of the number of iterations required vs. the acceleration parameter are given for a few cases in Figure 26. In a number of cases the optimum parameter was only 0.1 below the divergence limit for the particular case. A typical example of this appears in Figure 26b. The effect of the general field size is evident in Figure 26c and d, where the larger field (d) gives a much sharper minimum. Strong attraction with small fields makes the convergence more difficult as can be seen in Figures 26e and f. With strong attraction and a small field convergence was obtained only in a very narrow band of acceleration parameters.
A few trends are evident in these results:

(1) The optimum acceleration parameter, \( \omega^* \), and the number of iterations, \( I^* \), increase with the number of grid points.

(2) \( \omega^* \) increases slightly as the outer boundary moves outward, this effect being more pronounced at small outer boundary radius. \( I^* \) experiences a minimum as the outer boundary moves outward. Both of these effects are stronger with two bodies than with one.

(3) \( \omega^* \) is little affected by attraction amplitude with one body, but tends to decrease with increasing attraction amplitude with two bodies. (This difference is probably due to the fact that in the two-body case there was attraction to a line in the field, i.e., the cut between the bodies, as well as the body contours.) \( I^* \) tends to increase with increasing attraction.

(4) There is an optimum number of steps for addition of the attraction, with increasing \( I^* \) occurring on both sides of this optimum. Too few steps may produce divergence. This optimum is less apparent with two bodies than with one.

(5) The number of attraction lines had only a small effect on either \( \omega^* \) or \( I^* \) in the cases considered.

(6) \( \omega^* \) increases as convergence tolerance is tightened.

(7) The intermediate convergence tolerance value that should be used when the attraction is added gradually should be 0.01. A tighter tolerance requires more iterations, while a looser tolerance may not produce a sufficiently close initial guess for the next addition of attraction.

(8) The use of the variable acceleration parameter field is not generally recommended in cases where the optimum can be estimated from experience. This feature can be useful, however, in the absence of a good estimate. In that
case it is probably best to use the variable field once, and then to use a factor about 3% less than the average over the variable for subsequent runs.

IV. EXAMPLES OF COORDINATE SYSTEMS

A variety of results utilizing the theory and numerical procedures outlined in previous sections are now presented. The results selected for presentation here were chosen to exemplify the generality of the method, and some were used for the numerical fluids studies in Ref. 11. In most of the plots only a portion of the physical field is shown in order that the coordinate lines may be distinguishable near the bodies. The actual fields were generally extended some ten chords in radius from the bodies.

Coordinate System Control

As discussed in Section II, the curvilinear coordinate lines may be concentrated by attracting the lines to other lines or points in the field. The control of the coordinate system in this manner is illustrated in Figures 27-28. Input parameters involved are given in Table 11. In Figure 27, the basic system generated by the Laplace equations (zero right hand sides) is shown in (a). In (b) the η-lines have been attracted to the body. In (c) the attraction to the body has been made stronger on two sides, while in (d) the lines are more strongly attracted over a small portion of the body. In (e) and (f) the angle of intersection of the lines with the body has been controlled, over the entire body in (e) and over only a portion of the body in (f).

Figure 28 illustrates the use of control to pull the coordinate lines into a concave portion of the body contour, (a) being the result of the Laplace equations and (b) having the lines attracted to the slope discontinuity on the lower surface.
Various Body Shapes

Coordinate systems for a circular cylinder and a cambered Joukowski airfoil are pictured in Figure 29. The contours are of equi-spaced $\xi = \text{constant}$ and $\eta = \text{constant}$ lines in the physical plane. (In the interest of clarity only a portion of the grid is shown in this and subsequent figures. The outer boundary for these and all succeeding results is circular and has a radius of ten body-lengths.) Note that the two systems given in Figure 29 are orthogonal (The transformations in these instances are conformal.).

Slightly more general bodies are shown in Figure 30. These include a cambered and an integral flap Karman-Trefftz airfoil. Although systems for each of these could be generated by conformal transformations, the ones shown were obviously not. The effect of the coordinate system control is demonstrated for both airfoils in Figure 31. The figures exhibit the effect of contraction to the $\eta$-line coincident with the body profile. Note that the grid spacing is significantly collapsed in the contracted transformation. The contracted mesh spacing near the solid body boundary allows the solution of viscous flows (Ref. 11).

Contracted coordinate systems for more general airfoils are exhibited in Figures 32-34. These are the Liebeck laminar airfoil, the Göttingen 625 airfoil and a NACA 0018 profile. Contraction to the single-body $\eta$-line is demonstrated for the Liebeck profile and to the initial 15 $\eta$-lines for the Göttingen 625 and NACA 0018 airfoils. The effects of multiple-$\eta$-line contraction are seen to be quite dramatic.

Coordinate systems for a multiple-airfoil system are shown in Figure 35, with coordinate line concentration to the bodies and into the concave region formed by the airfoils and the cut between. The coordinate line attraction is to the first ten $\eta$-lines surrounding the bodies with an amplitude of 10,000
and a decay factor of 1.0 on all but the tenth line, where 0.5 was used. The coordinate system for simply connected regions are shown in Figures 36 and 37.

Finally, to demonstrate the applicability of the transformation method to quite arbitrary bodies, a system in contracted form (15 line) for a rather odd looking body—denoted the cambered rock—is given in Figure 38.

V. EXAMPLES OF APPLICATION TO PARTIAL DIFFERENTIAL EQUATIONS

As noted above, any set of partial differential equations may be solved on the boundary-fitted coordinate system by transforming the equations and associated boundary conditions and solving the transformed equations numerically in the transformed plane. All computation can be done on the fixed square grid in the rectangular transformed region regardless of the shape of the physical boundaries. Several examples of such application are given below.

Potential Flow (Ref. 11, 12, 19)

The two-dimensional irrotational flow about any number of bodies may be described by the Laplace equation for the stream function $\psi$:

$$\psi_{xx} + \psi_{yy} = 0 \quad (31)$$

with boundary conditions

$$\psi(x,y) = \psi_k \text{ on the surface of the } k\text{th body.} \quad (32a)$$

$$\psi(x,y) = y \cos \theta - x \sin \theta \text{ at infinity.} \quad (32b)$$

where $\theta$ is the angle of attack of the free stream relative to the positive $x$-axis. Here the stream function is nondimensionalized relative to the air-
foil chord and the free stream velocity. When transformed to the curvilinear coordinate system this equation becomes

\[ \alpha \psi_{\xi\xi} - 2B \psi_{\xi\eta} + \gamma \psi_{\eta\eta} + \sigma \psi_{\eta} + \tau \psi_{\xi} = 0 \]  

(33)

The transformed boundary conditions are (see Figure 1).

\[ \psi(\xi, \eta_1) = \psi_0 \text{ on } \eta = \eta_1 \text{ (i.e., on } \Gamma_1^*) \]  

(34a)

\[ \psi(\xi, \eta_2) = y(\xi, \eta_2) \cos \theta - x(\xi, \eta_2) \sin \theta \text{ on } \eta = \eta_2 \text{ (i.e., on } \Gamma_2^*) \]  

(34b)

The uniqueness is implied by insisting that the solution be periodic in \(-\infty < \xi < \infty, \eta_1 \leq \eta \leq \eta_2\). The coefficients \(\alpha, \beta, \gamma, \sigma, \) and \(\tau\) are calculated during the generation of the coordinate system (see Appendix A). Equation (33) was approximated using second-order, central differences for all derivatives, and the resulting difference equation was solved by accelerated Gauss-Seidel (SOR) iteration on the rectangular transformed field. The value of the boundary values of \(\psi\) on the bodies were determined by imposing the Kutta condition on each body.

The pressure coefficient at any point in the field may be obtained from the velocities via the Bernoulli equation, which in the present non-dimensional variables is

\[ C_p = 1 - \frac{|V|^2}{j_2^2} \]  

(35)

On the body surface this becomes

\[ C_p = 1 - \frac{y}{j_2} \psi_{\eta}^2 \]  

(36)
with the derivative evaluated by a second-order one-sided difference expression. The nondimensional force on the body is given by

$$ F = -\int C_p \eta \, ds $$  \hspace{1cm} (37)

where $\eta$ is the unit outward normal to the surface, and $ds$ is an increment of arc length along the surface. The lift and drag coefficients are

$$ C_L = \int C_p (-x_\zeta \cos \theta - y_\zeta \sin \theta) \, d\zeta $$  \hspace{1cm} (38a)

$$ C_D = \int C_p (y_\zeta \cos \theta + x_\zeta \sin \theta) \, d\zeta $$  \hspace{1cm} (38b)

These integrals were evaluated by numerical quadrature using the trapezoidal rule.

The coordinate system for a Karman-Trefftz airfoil having an integral flap is shown in Fig. 30b, and the streamlines and pressure distribution for this airfoil are compared with the analytic solution (Ref. 20) in Figure 39. Similar excellent comparisons have been obtained with other Karman-Trefftz airfoils. Fig. 32 shows the coordinate system for a Liebeck laminar airfoil, the solution for which is compared with experimental results (Ref. 21) for the pressure distribution in Fig. 40. Finally the coordinate system for a multi-element airfoil is shown in Fig. 35, with the streamlines and pressure distributions shown in Fig. 41. Here coordinate system control was employed as discussed above to attract the coordinate lines into the concave region formed by the intersections of the cut between the airfoils, as well as to the bodies.

Figure 42 shows a coordinate system for a pair of circular cylinders, the coordinate lines being attracted to the intersections of the cut between
the bodies with their surfaces. (The accompanying diagram shows the arrangement of the body and re-entrant segments in the transformed plane.) The streamlines for potential flow obtained on this coordinate system are shown in Figure 43a, and the surface pressure distribution is compared with the analytic solution (Ref. 22) in Figure 43b. By contrast, the uncontracted coordinate system, generated by Eq. (11), and resulting pressure distribution are shown in Figure 44. The effectiveness of the coordinate system control is clear in the comparison of these results with those in Figure 43b. To illustrate the use of different segment arrangements, Figures 45 and 46 show another coordinate system and pressure distribution for the same two cylinders.

The effects of the various numerical parameters involved for the potential flow solution were investigated in some detail, and the results are reported in Ref. 19. These results should serve as a guide to reasonable choices of such parameters as field size, convergence criteria, mesh spacing, etc. for the use of the body-fitted coordinate systems in other applications as well. Ref. 19 also serves to illustrate in some detail the procedure of application of the body-fitted coordinate system to the solution of a partial differential system.
Viscous Flow

The time-dependent, two-dimensional viscous incompressible flow about any number of bodies may be described by the Navier-Stokes equations in various formulations, two of which are illustrated below.

Vorticity-Stream Function Formulation. (Ref. 11-14) with the vorticity and stream function as dependent variables the transformed Navier-Stokes equations are

\[
\frac{\partial \omega}{\partial t} + \left( \psi \frac{\partial \omega}{\partial \xi} - \omega \frac{\partial \psi}{\partial \eta} \right) / J = (\alpha \omega \xi - 2 \beta \omega \eta) \]

\[
+ \gamma \omega \eta + \sigma \omega + \tau \omega \xi / J^2 R \tag{39}
\]

\[
\alpha \psi \xi - 2 \beta \psi \xi \eta + \gamma \psi \eta + \sigma \psi + \tau \psi \xi = -J^2 \omega \] \tag{40}

with boundary conditions:

\[
\psi = \text{constant}, \frac{\sqrt{Y}}{J} \psi \eta = 0 \text{ on body surface} \tag{41a}
\]

\[
\psi = y \cos \theta - x \sin \theta, \omega = 0 \text{ on remote boundary} \tag{41b}
\]

All quantities are non-dimensionalized with respect to the free stream velocity and the airfoil chord. All space derivatives in the field were represented by second-order, central difference expressions. The time derivatives were represented by two-point backward difference expressions. The \( \eta \)-derivatives on the body surface were represented by second-order one-sided difference expressions. The solution was implicit in time, all the difference equations being solved simultaneously by SOR iteration at each time step.

The boundary conditions were implemented directly except for the second of (41a), which was satisfied by adjusting the value of the vorticity on the body by a false-position iteration procedure until the second-order, one-sided difference representation of the tangential velocity, \( \frac{\sqrt{Y}}{J} \psi \eta \), was below some tolerance:
Here $k$ is the iteration count, $K$ an adjustable parameter, and $(i, l)$ refers to a point on the body surface.

The surface pressure is calculated from the line integral of the Navier-Stokes equations on the surface:

$$P_2 - P_1 = \frac{2}{R} \int_{\xi=1}^{\xi=2} \mathbf{\omega} \cdot d\mathbf{r} = \frac{2}{RJ} \int_{\xi=1}^{\xi=2} (\beta \omega_\xi - \gamma \omega_\eta) d\xi $$

(43)

The body force components are then obtained from the integration of the pressure and shear forces around the body surface:

$$F_x = + \dot{\phi} p y_\xi d\xi - \frac{2}{R} \dot{\phi} \omega x_\xi d\xi$$

$$F_y = - \dot{\phi} p x_\xi d\xi - \frac{2}{R} \dot{\phi} \omega y_\xi d\xi$$

(44a)

(44b)

Finally, the lift and drag coefficients are given by

$$C_L = \frac{F_y \sin \theta - F_x \sin \theta}{\frac{1}{2} \rho C^2}$$

$$C_D = \frac{F_y \sin \theta + F_x \cos \theta}{\frac{1}{2} \rho C^2}$$

(45a)

(45b)

where $\theta$ is the angle of attack.

The coordinate system for a Göttingen 625 airfoil shown in Fig. 33 was used in this solution. The high density of constant $\eta$-lines near the airfoil is the result of contraction to the first 15 $\eta$-lines. Streamline contours are shown in Fig. 47, and velocity profiles are shown in Fig. 48. Pressure and force coefficients are illustrated in Fig. 49.

To show that the boundary-fitted coordinate system can be used with arbitrary shaped bodies, the viscous flow about a cambered rock at a Reynolds number of 500 was developed. The contracted coordinate system used in the solution is given in Fig. 38. $\psi$ and $\omega$ contours are shown in

\[
\omega_{il}^{(k+1)} = \omega_{il}^{(k)} - K \left( \frac{\omega_{il}^{(k)} - \omega_{il}^{(k-1)}}{J/\psi_{il}^{(k)}} \right) \left( \frac{\sqrt{\gamma} \psi_{il}}{J/\psi_{il}^{(k)}} \right)
\]
Figure 50, and velocity profiles are shown in Figure 51.

In order that the pressure be single-valued, it is necessary that the value of the stream function on each body of a multiple-body system be such that the line integral of the Navier-Stokes equation on each body vanishes:

$$0 = \oint_{\partial B} \nabla \cdot p + \frac{1}{\rho} \nabla \cdot \frac{\nabla p}{2}$$

$$- \mathbf{v} \times \omega + \frac{1}{R} \nabla \times \omega \cdot d\mathbf{r}$$

$$= -\frac{1}{R} \oint_{\partial B} \left( \nabla \times \omega \right) \cdot d\mathbf{r}$$

Thus in the transformed plane it must be that

$$\oint_{\partial B} (\gamma \omega_{\eta} - \beta \omega_{\xi}) d\xi = 0$$

on each body. Since this requires a double iteration, i.e., for both $\omega$ and $\psi$ on each body, it appears that this formulation is not as well suited for two-body calculations as is the primitive variable formulation that follows.

**Velocity-Pressure Formulation** (Ref. 23, 14). With the velocity and pressure (primitive variables) as the dependent variables the transformed Navier-Stokes equations are

$$u_t + \left[ y_{\eta} (u^2)_{\xi} - y_{\xi} (u^2)_{\eta} \right]/J + \left[ x_{\eta} (uv)_{\xi} - x_{\xi} (uv)_{\eta} \right]/J$$

$$+ (y_{\eta} p_{\xi} - y_{\xi} p_{\eta} )/J = (\alpha u_{\xi \xi} - 2\beta u_{\xi \eta} + \gamma u_{\eta \eta}$$

$$+ \sigma u_{\eta} + \tau u_{\xi})/Rj^2$$

$$v_t + \left[ y_{\eta} (uv)_{\xi} - y_{\xi} (uv)_{\eta} \right]/J + \left[ x_{\eta} (v^2)_{\xi} - x_{\xi} (v^2)_{\eta} \right]/J$$

$$+ (x_{\eta} p_{\xi} - x_{\xi} p_{\eta})/J = (\alpha v_{\xi \xi} - 2\beta v_{\xi \eta} + \gamma v_{\eta \eta}$$

$$+ \sigma v_{\eta} + \tau v_{\xi})/Rj^2$$
\[ \alpha p_{\xi\xi} - 2\beta p_{\xi\eta} + \gamma p_{\eta\eta} + \sigma p_{\eta} + \tau p_{\xi} = - (y_{\xi u_{\xi}} - y_{\xi u_{\eta}})^2 \]

\[ - 2(x_{\xi u_{\eta}} - x_{\xi u_{\xi}})(y_{\xi v_{\xi}} - y_{\xi v_{\eta}}) \]

\[ - (x_{\xi v_{\eta}} - x_{\xi v_{\xi}})^2 - J^2 D_r \]

where

\[ D = \frac{(y_{\eta u_{\xi}} - y_{\eta u_{\eta}} + x_{\xi v_{\eta}} - x_{\xi v_{\xi}})}{J} \]

Equation (50) is the transformed Poisson equation for the pressure, obtained by taking the divergence of the Navier-Stokes equations.

The boundary conditions are

\[ u = v = 0 \text{ on body surface} \quad (52a) \]

\[ u = \cos \theta, \ v = \sin \theta, \ p = 0 \text{ on remote boundary} \quad (52b) \]

The pressure at each point on the body was adjusted at each iteration by an amount proportional to the velocity divergence evaluated using second-order one-sided differences for the \( \eta \)-derivatives on the body.

**Pressure Distribution and Force Coefficients.** The surface pressure distribution is calculated in the vorticity-stream function formulation from the line integral of the Navier-Stokes equation around the body surface. In the velocity-pressure formulation the surface pressure, is, of course, obtained directly. In the velocity-pressure formulation it is necessary to calculate the body vorticity before applying (44) from

\[ \omega = - \frac{1}{J}(y_{\xi v_{\eta}} + x_{\xi u_{\eta}}) \]

Figure 35 shows the coordinate system for a multiple airfoil consisting of two Karman-Trefftz airfoils, one simulating a separated flap. Coordinate system control was used to attract the coordinate lines strongly to the first ten lines around the bodies and to the intersections of the cut between the bodies with the trailing edge of the fore body and the leading edge of the aft body. Velocity vectors and pressure distributions for the viscous flow
solution at Reynolds number 1000 are shown in Figure 52.

Loaded Plate (Ref. 24)

Figure 53 shows a comparison between the numerical and analytic solution (Ref. 25) for deflection contours for a simply supported uniformly loaded triangular flat plate. This problem involves the solution of the biharmonic equation by splitting into two Poisson equations. The transformed Poisson equations appear as in Eq. (40) above. Again all computation was done in the rectangular transformed plane.
The coordinate system is generated by the program TOMCAT with the subroutines BNDRY, CORPLOT, LINWT, ERROR, GUESSA, MAXMIN, PARA, RHS, SOR, PLOT, and SYMBOL. Subroutines PLOT and SYMBOL were added for compatibility between the GOULD and CALCOMP plotters. Each facility will probably have to make minor changes for plotting. A complete set of instructions for the input is included in the listing of TOMCAT.

Core must be set to zero at load time.

Files. The program uses two essential files with internal names 10 and 11. File 11 is used to store a partially converged solution so that the iteration can be continued by a subsequent run. This file need not be retained once the solution has converged.

The converged coordinate system is written on file 10. This should be saved to use as input for PROGRAM FATCAT.
Certain files and additional control statements will also be necessary for the operation of the plotter, and these must be added to fit the user's installation. The program is compatible with both the GOULD and the CALCOMP plotters.

**Dimensions.** The standard program allows a maximum field size of 70 $\xi$ lines and 60 $\eta$ lines and requires a core size of 131,000 words for the Langley Research Center's CDC 6000 Series Computer System. Error signals and instructions for modification will be given if these limits are exceeded. The three statements requiring modification for larger fields are separated from the rest of the dimension and data statements of the main program for convenience.

**Input Parameters.** Most of the input is self-explanatory in the instructions given in the listing of TOMCAT. However, a few additional comments may be in order.

**Field Size.** The parameter IDISK controls the storage of the converged system on the disk file and also signals the restart of a partially converged solution. The format of the storage of the coordinate system on the disk file is given following IDISK in the instructions.

**Plotting.** Plotting may be by-passed by setting IPLOT to zero and eliminating certain control cards. The selection of the GOULD, CALCOMP, or other plotter is made by the parameter IPLTR. Recall that the user must also add certain site-dependent control statements to the run stream appropriate to the particular plotter installation. The parameters NUMBR and NUMBRI allow the
plotted field to be confined to a portion of the actual field by restricting
the number of curvilinear coordinate lines plotted. Coordinate lines may be
skipped in the plot by adjusting ISKIP1 and ISKIP2. The parameters XB1,
XB2, YB1, and YB2 also allow the plotted field to be restricted to the
portion of the actual field between limits in the cartesian coordinates.

Initial Guess. The parameter IGES controls the initial guess for the
iterative solution of the difference equations. Since these equations are
nonlinear, convergence can be obtained only from an initial guess within
some neighborhood of the solution. The same initial guess will not in
general give convergence for all segment arrangements. The type 0 is
suitable for single-body and simply-connected systems, however, as well as
for multi-body systems having all body segments on the same curvilinear
coordinate line. Types 2 and -40 are widely applicable to multiple-body
fields, except those having two bodies in single segments on opposite sides
of the transformed field, where types 1 or 4 may be effective. Very
strong coordinate attraction near a boundary having a sharp convex corner
requires an initial guess having sufficiently closely-spaced lines in the
region of line attraction, else the iterates may overlap the boundary. In
such a situation the exponential weighted average guess (type IGES > 4) should
be used. The lines in the guess will be more strongly contracted as IGES
is increased. Note that the use of this type of guess requires that the
boundary to which the lines are attracted be located on the bottom or left side
of the rectangular transformed field. Gradual movement of the outer boundary
may also help (see INFAC). See Section III for more information.

Body Contours. Points on the body and outer boundary contours may be placed
as desired around the contours, and the cartesian coordinates of these points
are input in order from cards, one card per point, or from a file with one
image per point.

Some of the contours of a multiple-body system may be split into several
segments which may be located on the rectangular boundary of the transformed field in many different ways as noted above in Section III. For single bodies the body and outer boundary contours are simply cut and opened into single segments. The points will be placed on the rectangular boundary from the first index, LB1, to the second index, LB2, even when LB1 exceeds LB2. The contour segments must be arranged so that the segment ends are connected by coordinate lines that do not cross. This is simply a matter of arranging the segments on the rectangular transformed field boundary such that a continuous path is traversed over the contour segments and connecting cuts in the physical field as a closed circuit is made of the rectangular boundary of the transformed field. The order in which these sets are input is immaterial, except that the outer boundary contour must be last. There is no relation between the order of input of the sets and the order of their appearance on the circuit of the rectangular boundary.

Note that no points are repeated in the input; the closure of each body contour is accomplished internally by the program. (The total number of $\xi$ and $\eta$ lines, IMAX and JMAX, however, will include the repeated points that close the contours. See, for example, the test cases given following the program listings in this section.

If a circular outer boundary is desired, this contour may be calculated internally rather than being input. In this case the radius and origin of the circle, and the angle of its initial point counter-clockwise with respect to the positive x-axis, are input. The points on the outer boundary contour will then be placed at equal angular increments clockwise from this initial angle. This outer boundary may then be located on the rectangular boundary in one or more segments in the same manner described above.

Re-entrant Boundaries. The re-entrant segments pairs are specified by their end points and the sides on which they lie, but no points are input
thereon, since these are actually cuts rather than boundaries in the phy-
sical plane. The order in which the re-entrant segments are input bears no
relation either to the order in which these segments occur on the circuit
of the rectangular boundary or to the order in which the body contours are
input.

**Acceleration Parameters.** If a non-zero value is input for R(1), then this
value will be used as a uniform SOR acceleration parameter. Typical values
for a number of cases have been given above in the Section III. The
program also has the capability of calculating a field of variable local
acceleration parameters which are updated at each iteration until the
maximum absolute change of acceleration parameter over the field is less
than the input value R(10), after which the acceleration parameter field is
frozen. Since these local parameters are calculated from linear theory, they
are not true optimum values. Furthermore, in certain local situations, not
even the linear optimum is known. A choice is given, via IEV, for these
situations, but under-relaxation is generally the safest course. As noted
in Section III, in some cases these calculated variable acceleration
parameters tend to be too high and may not give convergence. The use of
the variable acceleration parameters also requires extra computer time for
their calculation, of course, and this calculation involves a square root.
Therefore, the constant input acceleration parameter is usually to be
preferred, provided this value is selected with some care with attention to
the results in Section III.

**Coordinate System Control.** The curvilinear coordinate lines may be concen-
trated by attracting the lines to the body contours or other lines or to
points in the field. Generally an effective way for concentration in the
vicinity of a body contour is to use attraction to the contour and also to
the first several lines off the contour, with decreasing attraction amplitude
on each line outward and a decay factor of 0.5 or less on all lines.

On a field that is 10 chords in radius with 40 lines surrounding the
body, an attraction amplitude of 1000 with attraction to 10 lines gives
moderate concentration, while 100,000 gives very strong concentration.
Amplitudes of 100 or below give only slight changes from the concentration
that is inherent in the basic homogeneous equations.

Some attention must be paid to the rapidity of the change of coordinate
line spacing with strong attraction else truncation error in the form of
artificial diffusion may be introduced as follows: Consider the finite
difference approximation of a first derivative with variation only in the
x-direction to which the ξ-lines are normal. Then by (8)

\[ f_x = \frac{y_n f_{\xi}}{x_\xi y_n} = \frac{f_\xi}{x_\xi} \]

The difference approximation then would be

\[ f_x = \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} + T_i \]

where \( T_i \) is the local truncation error.

Taylor series expansions of \( f_{i+1} \) and \( f_{i-1} \) about \( f_i \) then yield, after
some algebraic rearrangement,

\[ T_i = -\frac{1}{2} (f_{xx})_i (x_{i+1} + x_{i-1} - 2x_i) \]

But the last factor is simply the difference approximations of \( x_\xi \), so that

\[ T = -\frac{1}{2} x_\xi f_{xx} \]

This truncation error thus introduces a numerical diffusive effect in
the difference approximation of first derivatives. Care must therefore be
taken that the second derivatives of the physical coordinates (i.e., the rate
of change of the physical spacing between curvilinear coordinate lines) are not
too large in regions where the dependent variables have significant second
derivatives in the direction normal to the closely spaced coordinate lines.
Just what is a permissible upper limit to the rate of change of the line spacing is problem dependent. Consider, for instance, viscous flow past a finite flat plate parallel to the x-direction. Here the velocity parallel to the wall changes rapidly from zero at the wall to its free stream value over a small distance that is of the order of \( \frac{1}{\sqrt{R}} \) where \( R \) is the Reynolds number, \( R = \frac{U_\infty x}{v} \), based on freestream velocity, \( U_\infty \), the distance from the leading edge of the plate, \( x \), and the kinematic viscosity, \( v \).

The equation for the time rate of change of the velocity parallel to the wall is

\[
\frac{\partial u}{\partial t} = -uu_x - v u_y + \frac{1}{R} (u_{xx} + u_{yy})
\]

Recalling that the large spatial variation in velocity occurs in the y-direction, coordinate lines would be contracted near the plate. The truncation error introduced by this contraction would be

\[
-v(-T) = (- \frac{v}{2} y_{\eta\eta}) u_{yy}
\]

This introduces a negative numerical viscosity \((- \frac{v}{2} y_{\eta\eta})\), since \( v \) and \( y_{\eta\eta} \) are both positive.

The effective viscosity is thus reduced (effective Reynolds number increased), so that the velocity gradient near the wall is steepened. Therefore care should be taken that \( y_{\eta\eta} \) is limited so that the numerical viscosity \((- \frac{v}{2} y_{\eta\eta})\) not significant in comparison with the physical viscosity \( \frac{1}{R} \).

The situation is mitigated somewhat of the fact that the numerical viscosity is proportioned to the small velocity normal to the wall, this velocity being of order \( \frac{1}{\sqrt{R}} \). Actually this limit is conservative, since the normal velocity drops to zero at the wall and only attains the order \( \frac{1}{\sqrt{R}} \) in the outer portion of the region of large gradient of velocity parallel to the wall where \( u_{yy} \) is very small.

Sufficiently close spacing of lines can be obtained even subject to such limits on the rate of change of the spacing by using decay factors in
the tenths range for the coordinate attraction.

The use of coordinate system control tends to slow the convergence of the iterative solution, and it is necessary to add the attraction gradually for strong concentrations. Convergence can be achieved even with very strong attraction amplitude by successively partially converging the field with a weaker amplitude and then using this result as the initial guess for the iteration with a stronger amplitude. This can all be done in one run by inputing the number of steps to be used for addition of the full amplitude (IFAC) and the multiple of the final convergence criterion to be used as the criterion for the partial convergence of each succeeding amplitude (EFAC). Generally the lowest or perhaps the next-to-the-lowest number of steps that will produce convergence is the most economical. In typical single-body fields, an amplitude of 1000 has required three steps, while 10,000 has required six steps. A value of 100.0 is typical for EFAC.

When very strong coordinate attraction is used to a boundary having a sharp convex corner the lines may tend to overlap the corner unless the SOR iterative sweep is toward this boundary. Since the sweep is done toward lower ξ and n values, such a boundary should be located on the bottom or left side of the rectangular transformed plane if strong attraction thereto is to be used. In such a case the initial guess should be IGES > 4 as mentioned above, the stronger the attraction, the larger IGES. When IGES is large enough it should not be necessary to use gradual addition of the attraction. Movement of the outer boundary may also help (INFAC).

Convergence of Very Large Fields. With some segment arrangements for multiple-body fields convergence problems have occurred with large fields (20 chords or so). This problem arises since with some arrangements, fewer lines pass between the bodies and the outer boundary in some directions than in others. Therefore, provision has been made for approaching con-
vergence on the final field by successively partially converging smaller fields and using each succeeding result to produce an initial guess by linear projection for the next larger field. This can be done in a single run by specifying INFAC and INFACO. A choice is given between doubling the field radius at each step and increasing it linearly. In the former case the initial size is completely determined by the number of steps specified, while in the latter case it is necessary also to specify the initial point in the linear increase from zero at which the radius is to start. Care should be exercised that the initial outer boundary does not intersect the bodies.
Coordinate System

Program TOMCAT
PROGRAM TOMCAT(INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT,TAPE10,TAPE11)

C ********** MISSISSIPPI STATE 2-D BODY-FITTED COORDINATE SYSTEM **********
C
C (DEPARTMENT OF AEROPHYSICS AND AEROSPACE ENGINEERING )
C ( MISSISSIPPI STATE UNIVERSITY 1975 )
C (DEVELOPMENT SPONSORED BY NASA,LANGLEY RESEARCH CENTER)

C DIRECT INQUIRIES TO DR. JOE F. THOMPSON
C DRIVER A
C MISSISSIPPI STATE, MS 39762
C PHONE 601-325-3625

C*********************************************************************
C
C DIMENSION X(70,60), Y(70,60), RETA(70,60), RXT(70,60), WACC(70,60)
C 1, TACC(70,60), XPLLOT(72), YPLLOT(72)
C DIMENSION R(13), IXER(2), IYER(2), CI(1), C2(1), BODY(6),
C      LBSID(6), LBI(6), LR2(6), LBDY(6), LR6TID(6), LR(6), LR2(6), LIDID(6), LI
C      21(6), LIZ(6), LTYPE(6), LSEX(6), INER(2)
C
C INTEGER TACC

C*****
C DATA NDIM,NDIM1 /70,60/
C DATA RADR/77,29077951308/
C DATA MNBEG,MNBEG /5,5/
C DATA ZERO /1,0E=08/

C*********************************************************************
C
C*********************************************************************
C*** CARDS(3) , C1/C2/BDY = FORMAT(BA10)
C (MAY BE BLANK)
C
C C1 AND C2 = 80 CHARACTER AN ARRAYS WHICH ARE PRINTED AT
C THE TOP OF EACH OUTPUT PAGE AND ON ANY PLOTS,
C
C BODY = NAME OF BODY BEING TRANSFORMED (80 CHARACTERS MAX).
C
C*** CARD I IMAX,JMAX,NBDY,ITER,IGEB,IOB8,ITGR,ITNTL,INFN,IGED
C = FORMAT(1015)
C
C IMAX = NUMBER OF XI-LINES,
C JMAX = NUMBER OF ETA-LINES,
C NBDY = NUMBER OF BODIES IN THE FIELD,
C (ZERO FOR SIMPLY-CONNECTED REGION)
C ITER = MAXIMUM NUMBER OF ITERATIONS ALLOWED.
C
C IGEB = INITIAL GUESS TYPE I (SEE IGED ALSO)
C (SOME SEGMENT ARRANGEMENTS WILL NOT CONVERGE
C WITH THE STANDARD INITIAL GUESS, THEREFORE
C SEVERAL ALTERNATIVES ARE PROVIDED)
C (IGE=0 IS TYPICAL FOR SINGLE-BODY FIELDS OR
C MULTIPLE-BODY FIELDS HAVING ALL BODIES ON THE
C SAME SIDE OF THE TRANSFORMED FIELD, IGEB=2 OR =40
IS TYPICAL FOR OTHER MULTIPLE-BODY FIELDS, EXCEPT
FOR THOSE HAVING TWO BODIES IN SINGLE SEGMENTS
ON OPPOSITE SIDES OF THE TRANSFORMED FIELD, IN THE
LATTER CASE TRY IGESE3 OR 4.
(IGESE3=4 IS MORE EFFECTIVE FOR CASES WITH THE
OUTER BOUNDARY ON THREE SIDES.)

1 = WEIGHTED AVERAGE OF FOUR PROJECTED
BOUNDARY VALUES.
0 = SAME AS 1 EXCEPT ZERO USED IN PLACE OF VALUES
ON CUTS.
2 = SAME AS 1 EXCEPT BOUNDARY VALUES ON CUTS
OMITTED IN AVERAGE.
3 = MOMENT PROJECTION
|XSUM*SUMX+SUMX)/SUMD, DIVIDED BY TOTAL
WHERE SUMS*SUM OF BOUNDARY VALUES,
SUMD*SUM OF DISTANCES TO BOUNDARY POINTS,
SUMX*SUM OF PRODUCTS OF ABOVE.
4 = SAME AS 3 EXCEPT NO DIVISION BY TOTAL
NUMBER OF BOUNDARY POINTS.
5 = SAME AS 2 EXCEPT EXPONENTIAL WEIGHT RATHER THAN
LINEAR CONCENTRATION TOWARD LOWER VALUES
OF XI AND/OR ETA WITH DECAY FACTOR OF
0.1*(IGESE4).
6 = EXPONENTIAL PROJECTION
X*SUMX*SUME, WITH EXPONENTIAL DECAY
FACTOR EQUAL TO EXP(-IGESE1),
WHERE SUMS*SUM OF EXP(-DECAYDISTANCE),
SUME*SUM OF PRODUCT OF BOUNDARY VALUE
AND ABOVE EXP.
(DISTANCE IS NONDIMENSIONALIZED RELATIVE
TO DIAGONAL OF RECTANGULAR TRANSFORMED FIELD)

1DISK = DISK READ/WRITE CONTROL.
0 = START ITERATION FROM INITIAL GUESS,
DON'T STORE COORDINATE SYSTEM ON DISK.
1 = START ITERATION FROM INITIAL GUESS,
STORE COORDINATE SYSTEM ON DISK.
2 = CONTINUE ITERATION OF A PARTIALLY
CONVERGED SOLUTION READ FROM RESTARTFILE
ON DISK, STORE COORDINATE SYSTEM ON DISK.
3 = AS #2 EXCEPT DON'T STORE COORDINATE SYSTEM.

NOTE: 1DISK OF 1 OR 2 CAUSES THE COORDINATE SYSTEM TO BE WRITTEN
TO DISK IN THE FOLLOWING FORMAT:

WRITE(10,1) C1
WRITE(10,1) C2
WRITE(10,1) IMAX,JMAX,NBSEG,NSSEG,LISEG,NBXY
WRITE(10,1) LB4ID,LB1(L),LB2(L),LBXY(L),LBEN(L),
1 LB1,NBSEG
WRITE(10,1) LRD(L),LID(L),LIX(L),LIB(L),LIS(L),LJ(L),LJK(L),
1 LTYPE(L),L=1,NSSEG
WRITE(10,1) ((X(I,J),I=1,IMAX,J=1,JMAX),
1 ((Y(I,J),I=1,IMAX,J=1,JMAX)
HERE LISSEG IS THE TOTAL NUMBER OF BODY SEGMENTS (EXCLUSIVE
OF OUTER BOUNDARY SEGMENTS) PLUS 1; LBEN(L) IS +1 IF

58
C ** LP2(L)>LR1(L) AND IS = 1 OTHERWISE LR1 AND LR2 ARE
INTERCHANGED INTERNALLY AFTER LSCH IS SET IF NECESSARY.
SO THAT LB2(LR1) LTYPE IS 1,2,3,4,5, OR 6, RESPECTIVELY.
IF THE TWO SEGMENTS OF A RE-ENTRANT PAIR ARE (1) ONE ON
BOTTOM AND ONE ON TOP, (2) BOTH ON BOTTOM, (3) BOTH ON TOP,
(4) ONE ON LEFT AND ONE ON RIGHT, (5) BOTH ON LEFT, OR
(6) BOTH ON RIGHT I X AND Y ARE THE CARTESIAN COORDINATES.

IF CONVERGENCE IS NOT ATTAINED IN THE ALLOWED NUMBER OF
ITERATIONS THE PARTIALLY CONVERGED SOLUTION IS STORED ON
DISK FOR RESTART, THE ITERATION MAY THEN BE CONTINUED BY
SETTING IDISK TO 2 OR 3 AND INCREASING ITER.

******

INIT = #0 DON'T PRINT EACH ITERATION ERROR NORM.
#1 PRINT EACH ITERATION ERROR NORM.

INIT = #0 DON'T PRINT INITIAL GUESS.
#1 PRINT INITIAL GUESS.

IMIN = #0 PRINT COORDINATE SYSTEM.
#1 DON'T PRINT COORDINATE SYSTEM.

IGED = CONTROLS DIRECTION OF WEIGHTED AVERAGES FOR INITIAL
GUESS TYPES 0,1,2, AND 4

= 0 = AVERAGE IN BOTH XI AND ETA DIRECTIONS
(FOR POINT AVERAGE)

= 1 = AVERAGE IN ONLY ETA DIRECTION
(TWO POINT AVERAGE)

= 2 = AVERAGE IN ONLY XI DIRECTION
(TWO POINT AVERAGE)

CARD 1 I PLOT, IPLTR, NCOPY, LIN1, LIN2, NUMBR, NUMBA1,
*** ISKIP1, ISKIP2, = FORMAT(915)

I PLOT = PLOT OPTIONS:
#0 NO PLOTS (INPUT OF REST OF CARD NOT REQUIRED).
#1 PLOT COORDINATE SYSTEM.
#2 PLOT INITIAL GUESS AND COORDINATE SYSTEM.

IPLTR = #1 PLOT WITH GOULD 4800,
#2 PLOT WITH GERMER
#3 OTHER (CALL PSEUDO DEVICE INDEPENDENT
VARIAIN AND CALCOMP DO NOT HAVE LINEWT
CAPABILITY)

NCOPY = NUMBER OF COPIES OF PLOT DESIRED.

LIN1 = PLOT LINE WEIGHT DESIRED FOR PLOT TITLES
(= 2, 1, 0, 2 RESPECTIVELY FOR TRIPLE, DOUBLE,
NORMAL, HALF, THIRD WEIGHT LINES)
(This applies only to the Gould 4800)

LIN2 = PLOT LINE WEIGHT DESIRED FOR COORDINATE SYSTEM,
(SEE NOTE BELOW LIN1)
(This applies only to the Gould 4800)

NUMBR = NUMBER OF ETA-LINES DESIRED FOR PLOT.
(ZERO VALUE PLOTS ALL)
NUMR1 = NUMBER OF XI-LINES DESIRED FOR PLOT,  
(ZERO VALUE PLOTS ALL)  
ISKIPI = SKIP PARAMETER FOR XI-LINES,  
(1 PLOTS EACH LINE, 2 PLOTS EVERY SECOND LINE, ETC)  
ISKIP2 = SKIP PARAMETER FOR ETA-LINES,  
(SEE NOTE BELOW ISKIPI)  
** EACH BODY AND THE OUTER BOUNDARY IS DIVIDED INTO ONE OR  
MORE SEGMENTS PLACED ON THE SIDES OF THE RECTANGULAR TRANSFORMED  
FIELD, THESE SEGMENTS ARE CONNECTED BY RE-ENTRANT SEGMENTS,  
THE CONFIGURATION INPUT IS AS FOLLOWS:  
** CARD I NRSEG, NRSEG = FORMAT(2I5)  
** NBSEG = TOTAL NUMBER OF BODY AND OUTER BOUNDARY SEGMENTS,  
** NRSEG = TOTAL NUMBER OF PAIRS OF RE-ENTRANT SEGMENTS,  
(ZERO FOR SIMPLY-CONNECTED REGION)  
** CARDS(NRSEG) = LBDY, LR1, LR2, LADY = FORMAT(4I5)  
** LBDY = SIDE OF RECTANGULAR TRANSFORMED FIELD ON WHICH  
BODY SEGMENT OR OUTER BOUNDARY SEGMENT IS LOCATED,  
(BOTTOM IS 1, LEFT IS 2, TOP IS 3, RIGHT IS 4,)  
** LR1, LR2 = FIRST AND LAST INDICES OF SEGMENT,  
(LR1 MAY EXCEED LR2)  
** LADY = OUTER NUMBER (BODIES ARE NUMBERED CONSECUTIVELY  
FROM 1 TO NBDY  
OUTER BOUNDARY IS BODY NUMBER 0 OR NBDY+1,  
NUMBER 0 CAUSES OUTER BOUNDARY TO BE CALCULATED  
INTERNALLY AS A CIRCLE, POSITIVE OR NEGATIVE  
NBDY+1 CAUSES OUTER BOUNDARY TO BE READ AS OTHER  
BODIES,.)  
** NOTE: EACH BODY IS READ VIA A SINGLE ADDR SET  
OF CARDS WITH NO REPEATED POINTS, NOT EVEN  
The first point, body segments must be  
CONSECUTIVE ACCORDING TO LBDY, WITH THE  
Outer boundary last regardless of its  
LADY, points in each segment are read  
CONSECUTIVELY AND PLACED ON BOUNDARY  
FROM INDEX LR1 TO INDEX LR2.  
** CARDS(NRSEG) = LRBD, LR1, LR2, LBDI, LII, LII = FORMAT(6I5)  
(Omit for Simply-Connected Region)  
** LRBD, LBDI = SIDES OF RECTANGULAR TRANSFORMED FIELD ON  
WHICH SEGMENTS OF A RE-ENTRANT PAIR ARE  
LOCATED, SEE NOTE BELOW LBDI FOR SIDES,  
(LBDI MUST EQUAL OR EXCEED LRBD)  
** LR1, LR2, LII = FIRST AND LAST INDICES OF EACH SEGMENT  
OF A RE-ENTRANT PAIR, (IF LBDI IS EQUAL  
TO LRBD THEN LII MUST EXCEED LR2, IF  
LBDI IS NOT EQUAL TO LRBD THEN LII  
MUST EQUAL LR1, IN ANY CASE LR2 MUST
### Card R(1), R(2), R(3), YINFIN, AINFIN, XOINF, YOINF, NINF =

**EXCEED L1, AND L12 MUST EXCEED L11.**

- **R(1)** = Gauss-Seidel acceleration parameter.
- **R(2)** = Convergence criterion for X iteration error norm.
- **R(3)** = Convergence criterion for Y iteration error norm.
- **YINFIN** = Radius of circular outer boundary.
- **AINFIN** = Angle of first point on outer boundary. (Degrees)
- **XOINF, YOINF** = Center of circular outer boundary.
- **NINF** = Number of unique points on outer boundary.

- **IEV**, **IAIT**, **R(10)** = FORMAT (215, F10.0)

**INFIN =** Number of unique points on outer boundary.

**INFAC**, **INFACO** = FORMAT (215)

- **IEV** = Controls complex Jacobi eigenvalue procedure.
- **IAIT** = Non-zero value causes variable acceleration parameter field to be printed.
- **R(10)** = Convergence criterion for variable acceleration parameter field. Field is frozen when maximum absolute change on field is less than R(10).

**INFAC** = Number of steps in attainment of outer boundary.

- Positive doubles outer boundary at each step.
- Negative moves outer boundary linearly.

**CARD** =

- **(Blank card may be input if constant acceleration parameter is used)**
- **(Can be used to aid convergence by converging a smaller initial guess for a larger field, blank card may be input if this feature is not to be used, standard is to not use this feature)**
**INCREASE MAGNITUDE IF DIVERGENCE OCCURS**

**INFAO** - INITIAL STEP IN ATTAINMENT OF OUTER BOUNDARY,
(LINEAR ATTAINMENT ONLY)

**CARD 1:** SIZE, RATIO, DIST, XR1, XB2, YR1, YR2 = FORMAT(7F10.0)
(Omit if no plots desired, i.e., if IPlot is zero)

**SIZE** = PLOT SIZE IN Y-DIRECTION, (INCHES)
(Typically 8.0)

**RATIO** = 0.0 X AND Y PLOT SCALES ARE EQUAL,
> 0.0 X AND Y PLOT SCALES ARE ADJUSTED SO THAT THE
PLOT IS SQUARE,
(Typically 0)

**DIST** = GAVD PAGES REQUESTED(X-DIRECTION), EITHER 10 OR 20,
(Typically 10.0)

**XR1, XB2** = MINIMUM AND MAXIMUM X-VALUES TO BE PLOTTED,
(ZEROB PLOT ALL)

**YR1, YR2** = MINIMUM AND MAXIMUM Y-VALUES TO BE PLOTTED,
(ZEROB PLOT ALL)

**AFTER THIS INPUT, READ IN BODY COORDINATES = FORMAT(2F10.0)**

**IF NO COORDINATE SYSTEM CONTROL IS TO BE USED, FOLLOW THESE CARDS**
**WITH THREE BLANK CARDS, IF CONTROL IS TO BE USED, USE THE**
**FOLLOWING INPUT RATHER THAN THE BLANK CARDS.**

**INPUT FOR COORDINATE SYSTEM CONTROL, USE TWO SETS, ONE FOR**
**XI-LINE CONTROL AND ONE FOR ETA-LINE CONTROL.**
**THIS DATA IS READ IN SUBROUTINE SOR.**

**CARD 1:** ATYP, ITYP, NLT, NPT, DEC, AMPFAC = FORMAT(A6, I4, 2I5, 2F10.0)

**ATYP** = TYPE OF ATTRACTION, (XI FOR XI-LINE ATTRACTION,
ETA FOR ETA-LINE ATTRACTION) LEFT JUSTIFIED,

**ITYP** = ZERO GIVES ATTRACTION ON BOTH SIDES,
NON-ZERO GIVES ATTRACTION ON CONVEX SIDE AND
REPULSION ON CONCAVE SIDE.

**NLT** = NUMBER OF ATTRACTION LINES.

**NPT** = NUMBER OF ATTRACTION POINTS.

**DEC** = NON-ZERO DEC USES DEC FOR DECAY FACTOR.

**AMPFAC** = NON-ZERO AMPFAC MULTIPLIES ALL AMPLITUDES
BY AMPFAC.

**CARDS(NLT)** = JLN, ALN, DLN = FORMAT(5X, I5, 2F10.0)
(Omit if NLT is zero)

**JLN** = ATTRACTION LINE INDEX.

**ALN** = AMPLITUDE (NEGATIVE REPELS) FOR LINE ATTRACTION.
**nL_ = DECAY FACTOR FOR LINE ATTRACTION.**

**CARDS(NPT) I IPT,JPT,APT,DPT = FORMAT(215,2F10,0)**

**ipt,Jpt,apT,dpt = ATTRACTION POINT INDICES.**

**APT = AMPLITUDE (NEGATIVE REPULS) FOR POINT ATTRACTION.**

**DPT = DECAY FACTOR FOR POINT ATTRACTION.**

**FOLLOW THE COORDINATE SYSTEM CONTROL CARDS WITH THE**

**FOLLOWING CARD: 361**

**CARD 1 IFAC,IRIT,EFAC = FORMAT(215,2F10,0)**

**IFAC = NUMBER OF STEPS IN ADDITION OF INHOMOGENEOUS TERM.**

**IRIT = NON-ZERO VALUE CAUSES INHOMOGENEOUS TERM TO BE PRINTED.**

**EFAC = MULTIPLE OF CONVERGENCE CRITERION TO BE USED FOR INTERMEDIATE CONVERGENCE BETWEEN ADDITIONS OF INHOMOGENEOUS TERM.**

**READ INPUT PARAMETERS**

**WRITE(6,640)**

**A CALL TO PSEUDO Initializes THE GRAPHIC OUTPUT SYSTEM. THIS MUST BE THE FIRST CALL FOR GRAPHICS IN ORDER TO GENERATE PLOT DATA IN THE FORM OF A DEVICE INDEPENDENT PLOT VECTOR FILE.**

**CALL TO LEROY SLOWS THE SPEED FOR GERBER FOR LIQUID INK PEN. IT IS OK TO LEAVE CALL LEROY IN FOR THE OTHER DEVICES.**

**CALL PSEUDO CALL LEROY READ (5,650) CI,C2,BOY READ (5,620) IMAX,JMAX,NDIM,ITER,IGEN,INDK,INIR,INTL,LOWIN,IGED IF (IMAX,GT,NDIM) WRITE (6,700) IMAX,NDIM IF (JMAX,GT,NDIM) WRITE (6,720) JMAX,NDIM**

**READ (5, 620) IMAX, JMAX, NDIM, OR, JMAX, GT, NDIM)**

**STOP 1**

**READ (5,620) IPLOT,IPLOT,NCOPY,INWT,INWT2,NUMBR,NUMBR1,I**

**18KIP1,18KIP2 READ (5,960) NBBEG, NRSEG IF (NBBEG,GT, NRSEG) WRITE (6,910) NBBEG, NRSEG IF (NRSEG,GT, NBBEG) WRITE (6,920) NRSEG, NBBEG IF (NBSEG,GT, NRSEG, OR, NRSEG, GT, NRSEG)**

**STOP 2**

**READ (5, 680) (LBBID(L),LB1(L),LM2(L),LADY(L),LB1, NBBEG)**
```c
IF (NRSEG.EQ.0) READ (5,890) (LHSIOL(L), L1(L), LR2(L), LISIOL(L), L1(L), L12(L), LME1, NRSEG)
READ (5,400) (R(I), IM1, IM2, YINFIN, AIMFIN, XOINF, YOINF, NINF)
READ (5,420) IEV, IAIT, R(10)
READ (5,620) INFAC, INFAC0
IF (I PLOT.NE.0) READ (5,630) SIZE, RATIO, DIST, XR1, XR2, YR1, YR2
C
WRITE INPUT PARAMETERS
C
*WRITE (6,1060)
WRITE (6,990) IMAX, JMAX, NRDY, ITRF, IGES, IMSH, IMIR, IMINL, INF
1, IGED
*WRITE (6,1000) I PLOT, IPLTH, MCPY, LMDIT, LMDI2, NUMB, NUMBR1
1, ISKIP1, ISKIP2
WRITE (6,1010) NRSEG, NRSEG
WRITE (6,1020) R(1), R(2), R(3), YINFIN, AIMFIN, XOINF, YOINF, NINF
WRITE (6,1030) IEV, IAIT, R(10)
WRITE (6,1040) INFAC, INFAC0
WRITE (6,1050) SIZE, PATT0, DIST, XR1, XR2, YR1, YR2
WRITE (6,1070)
IF (NRSEG.EQ.0) WRITE (6,1090)
WRITE (6,970) IGES
WRITE (6,990) (L, LHSIOL(L), LR1(L), LR2(L), LADY(L), LME1, NRSEG)
IF (NRSEG.EQ.0) WRITE (6,990) (L, LHSIOL(L), LR1(L), LR2(L), LISIOL(L), LME1,
111(L), L12(L), LME1, NRSEG)
IF (INFAC.EQ.0) INFAC=1
IF (INFAC.EQ.0) INFAC0=1
WRITE (6,950) YINFIN, AIMFIN, XOINF, YOINF, NINF, INFAC, INFAC0
IF (R(1).GT.0.0) WRITE (6,870)
C
SET UP PARAMETERS
C
NPLT8=1
AINFIN=AINFIN/RAD
IF (NUMBR.EQ.0) NUMBR=JMAX
IF (NUMBR1.EQ.0) NUMBR1=IMAX
IF (ISKIP1.EQ.0) ISKIP1=1
IF (ISKIP2.EQ.0) ISKIP2=1
IF (SRDY.EQ.0) SRDY=1
IF (I PLOT.EQ.0) GO TO 10
IF (ABS(XR1),LT,ZERO) XR1=1.0E20
IF (ABS(XR2),LT,ZERO) XR2=1.0E20
IF (ABS(YR1),LT,ZERO) YR1=1.0E20
IF (ABS(YR2),LT,ZERO) YR2=1.0E20
IF (DIST.LE.0.0) DIST=10.0
10 CONTINUE
C
READ POINTS ON BODIES AND
C
READ OR CALCULATE POINTS ON OUTER BOUNDARY
C
DO 20 L=1, NRSEG
20 IF (LABS.LEQ.LBDY(L).OR.LADY(L).EQ.0) GO TO 30
LISEG=NRSEG+1
GO TO 40
30 LISEG=1
40 DO 50 I=1, IMAX
   DO 50 J=1, JMAX
      X(I,J)=0.0
50   Y(I,J)=0.0
LADY=0
```
LBDY(NBSEG+1) = 100
DO 240 L = 1, NBSEG
   I1 = LBI(L)
   I2 = LHI2(L)
   ISEN = 1
   IF (I1 .GT. I2) ISEN = 1
   L8EN(L) = ISEN
   K1 = MINO(I1, I2)
   K2 = MAXO(I1, I2)
   LB1(L) = K1
   LB2(L) = K2
   IF (LBDY(L), EQ, LBDYO) I1 = I1 + ISEN
   IF (LBDY(L), NE, LBDY(L+1)) I2 = I2 - ISEN
   IF (LBDY(L), NE, LBDYO) I4 = I1
   K1 = MINO(I1, I2)
   K2 = MAXO(I1, I2)
   IGOTO = LBSID(L)
   GO TO (A0, 160, 60, 160), IGOTO
C**** BOTTOM OR TOP
   60 IF (LBSID(L), EQ, 1) J = 1
   IF (LBSID(L), EQ, 3) J = JMAX
   IF (NBSEG, EQ, 0) GO TO 90
   IF (L, LT, LISEG) GO TO 90
   IF (LBDY(L)) 90, 70, 90
   CALL BNDRY (X, Y, J1, J2, LBSID(L), YFIN, AFIN, XFIN, YINF, NINF, INDIM)
   GO TO 130
   90 DO 100 K = K1, K2
      I11 = (K = K1) * ISEN
      READ (5, 830) X(I1, J), Y(I1, J)
   100 IF (LBDY(L), EQ, LBDYO) GO TO 140
      I6 = J
      X56 = X(I4, I6)
      Y56 = Y(I4, I6)
      GO TO 150
   140 CONTINUE
      X(11, I4+J) = X35
      Y(11, J) = Y35
   150 IS = J
      IS12 = J
      X35 = X(13, IS12)
      Y35 = Y(13, IS12)
      IF (LBDY(L), EQ, LBDY(L+1)) GO TO 260
      IF (NBSEG, EQ, 0, AND, NBDY, GT, 1) GO TO 260
      X(I2+J, J) = X46
      Y(I2+J, J) = Y46
      GO TO 260
C**** LEFT OR RIGHT
   160 IF (LBSID(L), EQ, 2) I = 1
      IF (LBSID(L), EQ, 4) I = MAX
      IF (NBSEG, EQ, 0) GO TO 190
      IF (L, LT, LSEG) GO TO 190
      IF (LBDY(L)) 190, 170, 190
   170 CALL BNDRY (X, Y, J, I1, J2, LBSID(L), YFIN, AFIN, XFIN, YINF, NINF, INDIM)
      GO TO 230
   190 DO 200 K = K1, K2
      J11 = (K = K1) * ISEN
   200 READ (5, 830) X(I1, J), Y(I1, J)
   230 IF (LBDY(L), EQ, LBDYO) GO TO 240
CALL WAXMIN (X, T_AX, JMaXpNDI_, XB_AX, XTRTN, IDLwM, MlN, IDUM, IDUM, IDUM, 1, IDUM)

CALL MAXMIN (X, IMAX, JMAX, NDIM, XMAX, XRTN, IDUM, IDUM, 1, IDUM)

CONTINUE

LADY_1 = LADY (L)

DO 290 L = 1, NRSEG

290 LADY (L) = IABS (LADY (L))

C CLASSIFY RE-ENTRANT SEGMENTS

C IF (NRSEG EQ 0) GO TO 350

DO 340 L = 1, NRSEG

IGOTO = LBSID (L)

GO TO (300, 310, 320, 330), IGOTN

300 LTYPE (L) = 1

IF (LBSID (L) EQ LBSID (L)) LTYPE (L) = 2

GO TO 340

310 LTYPE (L) = 4

IF (LBSID (L) EQ LBSID (L)) LTYPE (L) = 5

GO TO 340

320 LTYPE (L) = 1

IF (LBSID (L) EQ LBSID (L)) LTYPE (L) = 3

GO TO 340

330 LTYPE (L) = 4

IF (LBSID (L) EQ LBSID (L)) LTYPE (L) = 6

340 CONTINUE

350 CONTINUE

C PULL OUTER BOUNDARY IN FOR INITIAL GUESS IF INFAC NOT ZERO

C IF (INFAC, GT, 0) CINFAC = 1.0/FLOAT (2** (1-INFAC))

C IF (INFAC, LT, 0) CINFAC = FLOAT (INFAC)/ABS (FLOAT (INFAC))

C IF (LRSEG, GT, NRSEG) GO TO 410

DO 400 L = LSEG, NRSEG

11 = LB1 (L)

12 = LB2 (L)

IGOTO = LBSID (L)

GO TO (360, 380, 360, 380), IGOTO

360 IF (IGOTO, EQ, 1) J = 1

IF (IGOTO, EQ, 3) J = JMAX

DO 370 I = 1, 12

X (1, J) = X (I, J) * CINFAC
Y(I,J) = Y(I,J) * CINFAC

GO TO 400

IF (IGOTO,EQ,2) IM1
IF (IGOTO,EQ,4) IMAX

DO 390 J = 1, IMAX
   X(I,J) = X(I,J) * CINFAC

390 Y(I,J) = Y(I,J) * CINFAC

GO TO 400

C INITIAL GUESS

CALL GUESS (IMAX, JMAX, NDOIM, Y, YMAX, YRMIN, YRMAX, YRMIN, YRINF, NRS)
   EIG, LPI, LP2, L11, L12, LB1, LB2, LIBID, LIES, IGES, IGEN)

C PRINT INITIAL DATA

WRITE (6, 640)
   WRITE (6, 640) C1
   WRITE (6, 640) C2
   WRITE (6, 670) NBID, YMAX, IMAX, R(1), R(2), R(3)
   WRITE (6, 660) NBID
   IF (NBID, EQ, 1) GO TO 470

490 IF (IPLOT, EQ, 2) CALL CORPLT (X, Y, NDOIM, NNUMAX, NUMR, C1, C2, RATIO, BIZ
   NCOPY, LINWT1, LINWT2, ISKIP1, ISKIP2, XPLT1, XPLT2, DNST, IPLT1, IPLT2, 
   LBI1, LB2, LBH, LR81, LR82, L11, L12, LTYP, NBSEG, NRS)
   RSEG, LISEG, JMAX, IMAX, XH1, XH2, YR1, YR2)

C PRINT FINAL VALUES

IF (IMFIN, NE, 0) GO TO 570
   WRITE (6, 670)
   WRITE (6, 670) C1
   WRITE (6, 670) C2
   WRITE (6, 670) R(1)
   WRITE (6, 670) YMAX
   WRITE (6, 670) JMAX
   WRITE (6, 670) R(2)
   WRITE (6, 670) R(3)
   WRITE (6, 670) IGES
   WRITE (6, 670) IGEN
   WRITE (6, 670) NRS
   WRITE (6, 670) NBID
   WRITE (6, 670) NUMR
   WRITE (6, 670) C1
   WRITE (6, 670) C2
   WRITE (6, 670) R(1)
   WRITE (6, 670) R(2)
   WRITE (6, 670) R(3)

570 WRITE (6, 670)

GO TO 580

580 WRITE (6, 670) ITER
68

530

550

S_O
570

,_ITE

b_1
662
t665

{6,8nnl

_QITE
(6,_50)
TXERCI),TXFR(2)eIYF_(1),IVF_(_)
*mITE
(_,7_0)
_0 550
JII,JMiX
,mITE
(_,75o)
J
_ITE
(6,71n)
Cx(Y,J)oI¢IoIMaX)
*_ITE
(6e7_0)
Dn 560
JsleJ_X
_RITE
(b_75_)
J
,_ITE
(6,71e)
(Y(I.J).ImI.I_AX)
_RITE
(6,6Q0)
TK (ZDISK.EU.O.n_.TDISK.EQ._)
_P TO 6_
_RITE
(10,q80)
C1
wRITE
(lO,gSO)
C2
TF (IDISK.EQ.,)
GO TO $80
*RITE(IO_620)
I_AX,J_X,_aSEG,_'_$£G,L?SEG,_BDY
_RITE(IO.qFI)TLSBT_CL),L_](L),LR2(L),LBnY_L_.LSE_rL),Lmt,N_SEG)
wRITE(IO.ga_)(LRST_(L),L_I(L),L_?(L),LIST_(L),LIIfLI,LI2(1),

b6_
6e7

67n
_71
e73
675

678
67_

tLTYPE(L),LBI,N@$EG)
GO TO _0
580
5_

_RITE(IO,_I)
IMAX,J"_X
*_I?E(10eg83)[(XC_,l),Im1,I_AX),Jm1,J_iX),(CY(led)_lileI_al)_Jlle
tj_X)
600
COnTInUE
PLOT
IF
(IPLOT.GT.O)
STOP
610

_0

Tn

68U
bBb
68T
668
68_

61n

CALL
CORPLT
(X,Y,NDI_,4tJ_RR1,NU"BR,C1,C_,RATIO,$IZE,NCOPY_LINWT1,L
IIN_T_,ISKIPI,IB_IP_eXPLOTeYPLOT,DTST_TPLT_NPLT$,LBSIDeLBle
_LB_,LSOYeL_$ID,LISIO,L_t,LH_,LII,LI_eLTYPEe_BBEG,NRSEGeLI$EG,JMAXe
31_AX_X_t_XR_,Y_I_YR_)
IF
(_PLOT.GT._)
CALL
PLOT
(O..O..g09)
STOP
010t

6ZO
6]0
b_O
6S0
660
&TO

FORMAT
(1615)
FORMAl
{8F10.0)
FORMAT
_IH1)
FORMAT(BA|O)
FORMAT(_tXeSAtO)
FORMAT
{/]TXe*BODY_FITTED
COORDINATE
SY_T_M*tt_IX,*TRA_FOR_ED
80D
IY.
*,8AtOIt_IX_*FtELD
PARAMETERS,
NU_RER
OF XI=LINE8
sw,Igt3RXe*
N
_UMBER
OR ETA-LINES
• *,I_t/ItqX,*
ITF_ATION
PARAMETEmS!
$OR ACCEL
]ERATION
PARAMETER
• *,FS,St_ZX,*
MAXI_u_
NUMBER
OF ITERATIONS
ALLO
_wEO
u *,I=/3_X,*
ALLOWABLE
ITERATIO_
ERROR
NORMSl
Xl
*,E10,_/75
SX_*
Yl
*,E|O,S)
bBO FORMAT
{I_IX_*NO
PLOTS
DESIRED*)
650 FORMAT
(/EIX,*PLOT
PARAMETERS1
COPIES
DESIRED
• *I3/_?x,*
LINE*,
1*wEIGHT
_ES_RED
• *I3t_qxe*PLOT
SIZE
TN YoDIRECTIDN
• *F§,]/3qX_*R
|ATlO
I *_F8,$)
FORMAT
(*0--ERROR ---IMAX
TO0 LARGE*,IOX,*IMAXs*,TS_SXn*MAXI_U
tM IB*,IS/t6X,*INCREASF
NDIM
I N OATA STATE.ENT
AND*,*
FIR8T
DI_ENSl
_ON OP X,V_RETA,RXI_ACC,TACC_*/16X,eALSO*,*
INCREASE
nIMENSTON
OF
SXPLOT
AND
YPLOT
TO MAXIMUM
OF*,*
NDIM
AND NDIMI
PtUS
_,*)
710
FORMAT
(_Xe10F11.S)
?_0
FORMAT
(*OmoERROR
..o=
JMAX
TOO LARGEteIOXetJMAXi*e_eSXeeMAXtMU
IM IS,,I_/t6X,*INCREASE
NDIMt
TN DATA
STATEMENT
AND*,,'
SECOND
DIMEN

69t
69_
6q3
69_
b_6
6_7
698
bq9
7O0
701

TO_
?Oh
_OT
?08
710
T11
Tt|
Tl$

?00

TiT
710
715


28ION OF X,Y,RETA,RXI,*ACC,TACC,*/16X,*ALSN*,* INCREASE DIMENSION O
3 YPLOT AND YPLOT TO MAXIMUM OF*,* NOIM AND NOIM1 PLUS 2,*
73 FORMAT (/21X,*INITIAL GUESSES FOR X AND Y*)
740 FORMAT (/5X,33(1H*),9X,ARRAY*8X,53(1H*))
750 FORMAT (5X,*INIT*)
760 FORMAT (/5X,33(1H*),9X,ARRAY*8X,53(1H*))
770 FORMAT (/5IX,*FINAL VALUES*/) 
780 FORMAT (21X,*ITERATION CONVERGES*) 
790 FORMAT (21X,*ITERATION IS CONVERGING BUT DOES NOT REACH ERROR TOL
1RANCES IN *13,* ITERATIONS*)
800 FORMAT (21X,*ITERATION CONVERGES*)
810 FORMAT (/21X,*INITIAL ITERATION ERROR NORMS X I *,E10.5*,Y I *,E10.5*,
110.5*,AT ITERATE # 1/21X,*FINAL ITERATION ERROR*, NORMS X 1
2*,E10.5*,Y I *,E10.5*, AT ITERATE #*,I4)
820 FORMAT (215,F10.0)
830 FORMAT (2F10.0)
850 FORMAT (21X,*LOCATION OF MAXIMUM ITERATION ERROR; X I I*,I5*,I*,J I I*,I5*)
1/58X,*Y I I*,I5*, J I I*,I5*)
860 FORMAT (/21X,*NUMBER OF ROBIES IN FIELD I I*)
870 FORMAT (*UNIFORM ACCELERATION PARAMETER USED,*)
880 FORMAT (2F10.5)
890 FORMAT (61S)
900 FORMAT (7F10.0,215)
910 FORMAT (*==ERROR NBBEG TOO LARGE*,10X,*NBBEG**,I3,5X,MAXI
1MUM 18*,I3/16X,*INCREASE MNRSEG IN DATA STATEMENT AND*,* DIMENSION
28 OF LBBID,I1,LB1,LBBID,LSN,*)
920 FORMAT (*==ERROR NBBEG TOO LARGE*,10X,*NBBEG**,I3,5X,MAXI
1MUM 18*,I3/16X,*INCREASE MNRSEG IN DATA STATEMENT AND*,* DIMENSION
28 OF LBBID,L1,L2,LBBID,LI1,L12,LTYPE,*)
930 FORMAT (*==BODY SEGMENTS==*/3X,*L*2X,LBBID,4X,LB1,4X,LB2,4X
13X,LBDY*/(I4,4I7))
940 FORMAT (*==RE-ENTRANT SEGMENTS==*/3X,*L*2X,LBBID,4X,L1,4X,L2,4X
13X,LBDY*/(I4,4I7))
950 FORMAT (*==OUTER BOUNDARY==*/3X,*RADIUS**,F12.8,10X,*INITIAL AN
GLE *F13.8,3X,*ORIGIN AT X **F11.8*, Y **F11.8/3X,*NUMBER
20F POINTS **I4,10X,15,* STEPS IN*,* ATTAINMENT OF INFINITY*,10X,*
3INITIAL STEP (LINEAR CASE) **I4)
960 FORMAT (215,F10.0)
970 FORMAT (*0INITIAL GUESS TYPE1*,I4)
980 FORMAT (8A10)
981 FORMAT (5I5)
982 FORMAT (7I5)
983 FORMAT (8E10.2)
990 FORMAT (*0IMAX,*JMAX,NBDY,ITER,IGES,1DIBK,1KIN,1WINT,1FINF,IGED 1*
14,10I5)
1000 FORMAT (*0IPLOT,1PLTR,1NCPY,1LINWT1,1LINWT2,1NUMR,1NUMR1,1*
1I1BK1,1BK12,4I5)
1010 FORMAT (*0NBBEG,NBBEG*,215)
1020 FORMAT (*0R(1),R(2),R(3),1FINF,AINFIN,1KIN,1FINF,1KINF,1FINF,1KINF*
15,8,1I5)
1030 FORMAT (*0IEV,1A1T,R(10) 1*,215,F15.8)
1040 FORMAT (*0INMP,IFNCO 1*,215)
1050 FORMAT (*0BIZE,RATIO,1DIBK,XB1,XR2,YB1,YB2*,7F12.8)
1060 FORMAT (1X,17(1H*),* INPUT *,3O(1H*))
1070 FORMAT (1X,13O(1H*))
1080 FORMAT (*0SIMPLY CONNECTED REGION*)

END
SUBROUTINE BNDRY (X,Y,IJ,II,IBID,R,A,XO,YO,INF,N)

C ******************************** CIRCULAR OUTER BOUNDARY ********************************
C *
C ** THIS ROUTINE CALCULATES NINF X,Y COORDINATES AROUND A CIRCLE OF
C ** RADIUS R AT EQUALLY SPACED ANGULAR INCREMENTS STARTING AT ANGLE A
C ** (POSITIVE COUNTER-CLOCKWISE FROM POSITIVE X-AXIS) AND PROCEEDING
C ** COUNTER-CLOCKWISE FROM THIS ANGLE.
C *
C******************************************************************************
C
DIMENSION X(N,1), Y(N,1)
DATA PI/3.14159265359/;
C*****
K1=MIN0(II,II+1)
K2=MAX0(II,II-1)
IBEN=1
IF (II,GT,II+1) IBEN=-1
IIXM1=INF
DB=2.0*PI/FLOAT(IIXM1)
GO TO (10,30,10,50), LSID
C**** BOTTOM OR TOP
10 DO 20 K=K1,K2
   IM=II+(K=K1)*IBEN
   X(I,I)=R*COS(A)*X0
   Y(I,I)=R*SIN(A)*Y0
   A=0
   GO TO 20
20  A=A+D
   GO TO 50
C**** LEFT OR RIGHT
30 DO 40 K=K1,K2
   JM=II+(K=K1)*IBEN
   X(I,J)=R*COS(A)*X0
   Y(I,J)=R*SIN(A)*Y0
   J=0
   GO TO 40
40  J=J+D
   CONTINUE
RETURN
C
END

SUBROUTINE CORPLT (X,Y,NDIM,NUMB1,NUMB2,C1,C2,RATIO,BIZE,NCOPY,L1
INW1,LINW2,ISPifi1,ISPfi2,XPLOT,YPLOT,DIST,PLOTE,NPLT8,LBB1
20,LBI,LBDY,LRBID,LIBID,LR1,LR2,LI1,LI2,LTYPE,NBSEG,NRSEG,LIBEG
3,JJMAX,II1MAX,X81,X82,Y81,Y82)

C *** PLOT ROUTINE - PLOTS COORDINATE SYSTEM AND SEGMENT DIAGRAM ******
C *
C******************************************************************************
C
DIMENSION LBID(1), LBI(1), LBDY(1), LRBID(1), LIBID(1), L
P1(1), LR2(1), LI1(1), L11(1), LTYPE(1)
DIMENSION X(NDIM,1), Y(NDIM,1), XPLOT(1), YPLOT(1), C1(1), C2(1)
1AXISL(2), XA(6), YA(6)
DATA SCAL /0,1/
DATA 81 /0,0,0875/
DATA 82 /0,175/
DATA H1 /1,0/
DATA H9 /0,5/

C*****
ISY=27
JMAX=NUMBR
IMAX=NUMBR!
H1=0.5*81
H2=1.5*81
H3=0.5*62
H4=1.5*62
H5=0.5*81
H6=2.0*81
H7=1.5*82
IMAX1*IMAX=1

C AXIS MINIMUM AND SCALE FACTORS

CALL MAXMIN (X,IMAX,JMAX,NDIM,XMAX,XMIN,I壹M,XMMX,IXMN,JXMN,ISKIPI
   1,ISKIPI2)
CALL MAXMIN (Y,IMAX,JMAX,NDIM,YMAX,YMIN,IXMx,JXMx,IXMN,JXMN,ISKIPI
   1,ISKIPI2)
XMAX=A M X I (XMAX,XYR2)
XMIN=A M X I (XMIN,XYR1)
YMAX=A M X I (YMAX,YYR2)
YMIN=A M X I (YMIN,YYR1)
X1=XMIN
Y1=YMIN
AXISL(2)=SIZE
IF (RATIO) 10,10,20
10 Y2=(YMAX-YMIN)/AXISL(2)
X2=Y2
AXISL(1)=(XMAX-XMIN)/X2
GO TO 30
20 AXISL(1)=AXISL(2)
X2=(XMAX-XMIN)/AXISL(1)
Y2=(YMAX-YMIN)/AXISL(2)

C SET UP PLO TTER

30 CONTINUE
NPLOT#2

C LABELS

IF(IPLTR.EQ.1.OR.IPLTR.EQ.2) CALL LINWT(IPLTR,LINWT1)
CALL PLOT (.5,.2,.3)
CALL SYMBOL (.0,.001,.0875,.0675,.1,.9,.80)
CALL SYMBOL (.3,.001,.0875,.2,.9,.80)
CALL PLOT (.1,.6,.3)
CALL PLOT (.0,.9,.99,.2)
CALL PLOT (.5,.5,.3)

C PLOT LINES OF CONSTANT ETA

IF(IPLTR.EQ.1.OR.IPLTR.EQ.2) CALL LINWT(IPLTR,LINWT2)
DO 50 J=1,JMAX,ISKIPI
   K=0
   DO 40 I=1,IMAX,ISKIPI
      IF (X(I,J),GT,XB2,OR,X(I,J),LT,XB1) GO TO 40
      IF (Y(I,J),GT,YB2,OR,Y(I,J),LT,YB1) GO TO 40
   40 CONTINUE
   K=K+1
50 CONTINUE
**K** = K + 1
XPLT(K) = X(I, J)
YPLOT(K) = Y(I, J)

CONTINUE
XPLT(K + 1) = X1
XPLT(K + 2) = X2
YPLOT(K + 1) = Y1
YPLOT(K + 2) = Y2

50 CALL LINE (XPLT, YPLOT, K, 1, 0, 0, 07)

**PLOT LINES OF CONSTANT XI**

ON 70 IMAX, ISKIP1
K = 0
DO 60 J = 1, JMAX, ISKIP2
IF (X(I, J) GT XBE OR X(I, J) LT XBE) GO TO 60
IF (Y(I, J) GT YBE OR Y(I, J) LT YBE) GO TO 60
K = K + 1
XPLT(K) = X(I, J)
YPLOT(K) = Y(I, J)

CONTINUE
XPLT(K + 1) = X1
XPLT(K + 2) = X2
YPLOT(K + 1) = Y1
YPLOT(K + 2) = Y2

70 CALL LINE (XPLT, YPLOT, K, 1, 0, 0, 07)

**SEGMENT CONFIGURATION DIAGRAM**

CALL PLOT (AXISL(1) + 2, 0, 1, 0, 3)
JMAX = JMAX
IMAX = IMAX

**BODY SEGMENTS ****

DO 100 LB = 1, NBSEG
D1 = LB1(L) * SCAL
D2 = LB2(L) * SCAL
IGOTO = LBID(L)
GO TO (80, 90, 80, 90), IGOTO
80 IF (LBID(L), EQ, 1) D3 = SCAL

**BOTTOM OR TOP**
IF (LBID(L), EQ, 3) D3 = JMAX = SCAL
LWT = 2
IF (LBID(L), EQ, 3) LWT = 1
IF (IPLTR, EQ, 1, OR, IPLTR, EQ, 2) CALL LINKT(IPLTR, LWT)
CALL PLOT (D1, D3, 3)
CALL PLOT (D2, D3, 2)

IF (IPLTR, EQ, 1, OR, IPLTR, EQ, 2) CALL LINKT(IPLTR, 0)
IF (LBID(L), EQ, 1) LB = 2
IF (LBID(L), EQ, 3) LB = 3
MM = SIGN(H, H) = S2 * (1.0 + SIGN(1, 0, M)) * 0.5
MM = SIGN(H, H)
CALL NUMBER (D1 = 1, D3 = H, 81, FLOAT(LB1(L)), 0, 0, 1)
CALL NUMBER (D2 = 1, D3 = H, 81, FLOAT(LB2(L)), 0, 0, 1)
IF (IPLTR, EQ, 1, OR, IPLTR, EQ, 2) CALL LINKT(IPLTR, 1)
IF (L, LBID(L), EQ, 0) CALL NUMBER ((D1 + D2) * 0.5 = HS, D3 = H, 82, FLOAT(LBDY(I)), 0, 0, 1)
IF (IPLTR, EQ, 1, OR, IPLTR, EQ, 2) CALL LINKT(IPLTR, 0)
CALL PLOT (D1,D3,H,H,3)
CALL PLOT (D1,D3,H,H,2)
CALL PLOT (D2,D3,H,H,3)
CALL PLOT (D2,D3,H,H,2)
GO TO 100
C**** LEFT OR RIGHT
90 IF (LRBID(L),EQ,2) D3=SCAL
   IF (LRBID(L),EQ,4) D3=JMAX*SCAL
   LWT=2
   IF (L,GE,LIBEG) LWT=1
   IF (IPLTR,EQ,1,OR,IPLTR,EQ,2) CALL LINAT(IPLTR,LWT)
   CALL PLOT (D3,D1,3)
   CALL PLOT (D3,D2,2)
   IF (IPLTR,EQ,1,OR,IPLTR,EQ,2) CALL LINAT(IPLTR,0)
   IF (LRBID(L),EQ,2) H=H6
   IF (LRBID(L),EQ,4) H=H6=81
   H1=SIGN(H7,H)=82*(1.0+SIGN(1,0)*H)*0.5
   H=8*SIGN(H5,H)
   CALL NUMBER (D3,H,H1,81,FLOAT(LR1(L1)),0.0,0,=1)
   CALL NUMBER (D3,H,H1,81,FLOAT(LR2(L1)),0.0,0,=1)
   IF (IPLTR,EQ,1,OR,IPLTR,EQ,2) CALL LINAT(IPLTR,0)
   IF (L,LT,LIBEG) CALL NUMBER (D3,H,H,0.0,0.5,82,FLOAT(LRB1(L))
1/0,0,0,=1)
   IF (L,LT,LIBEG) CALL SYMOL((D+H,H),(D+D2)*0.5,82,15Y,0,0,=1)
   IF (IPLTR,EQ,1,OR,IPLTR,EQ,2) CALL LINAT(IPLTR,0)
   CALL PLOT (D3+H,H,D1,3)
   CALL PLOT (D3+H,H,D1,2)
   CALL PLOT (D3+H,H,D2,3)
   CALL PLOT (D3+H,H,D2,2)
GO TO 100
C
C**** REENTRANT SEGMENTS ****
C
IF (NRBEG,EQ,0) GO TO 260
DO 250 LR1=1,NRBEG
   D1=LR1(L)*SCAL
   D2=LR2(L)*SCAL
   IF (IPLTR,EQ,1,OR,IPLTR,EQ,2) CALL LINAT(IPLTR,0)
   IGOTO=LBID(L)
   GO TO (110,120,110,120), IGOTO
250 CONTINUE
C**** BOTTOM OR TOP = FIRST OF PAIR
110 IF (LRBID(L),EQ,1) N3=SCAL
   IF (LRBID(L),EQ,3) N3=JMAX*SCAL
   CALL PLOT (D1,D3,3)
   CALL PLOT (D2,D3,2)
   GO TO 130
C**** LEFT OR RIGHT = FIRST OF PAIR
120 IF (LRBID(L),EQ,2) D3=SCAL
   IF (LRBID(L),EQ,4) D3=JMAX*SCAL
   CALL PLOT (D3,D1,3)
   CALL PLOT (D3,D2,2)
130 D=Li1(L)*SCAL
   D=Li2(L)*SCAL
   IGOTO=LBID(L)
   GO TO (140,150,140,150), IGOTO
C**** BOTTOM OR TOP = SECOND OF PAIR
140 IF (LHID(L),EQ,1) D=SCAL
   IF (LHID(L),EQ,3) D=JMAX*SCAL
   CALL PLOT (D4,D6,3)
   CALL PLOT (D5,D6,2)
**SUBROUTINE LINWT(IPLTR, IPEN)**

```
*************** SETS PLOT LINE WEIGHT ***************
```

```
GO TO (10, 20), IPLTR
10 CALL LINWT(IPEN)
RETURN
20 CALL PENBL(IPEN)
RETURN
END
```
FUNCTION ERROR (VNE,HOLD,FLD,FILE,IVR)
DIMENSION IVER(1)

C

************** MAXIMUM NORM OF ITERATE CHANGE **************
C *

C

initial guess *************
C

C

************** INITIAL GUESS **************
C *

C

************** PROJECTION FROM BODY SEGMENTS
C

IF (IGES,NE,0) GO TO 30
T1=0.1*(IGES=4)
ARG2=T1*YXIM1
T2=1.0/(EXP(ARG2)-1.0)
ARG3=T1*YXIM1
T3=1.0/(EXP(ARG3)-1.0)
DFAC=1.0
DFACC=1.0
IF (IGED,EQ,1) DFACC=0.0
IF (IGED,EQ,2) DFACC=0.0
DO 20 J=2,10

C

10 IF (IGES,GE,5) GO TO 8
FACC=FLOAT(J=1)/FLOAT(YXIM1)
GOTO 2
6 ARG1=1.0*(J=1)
FACC=EXP(ARG1)=1.0)*T2
CONTINUE
DO 10 I=2,10

C
GO TO 9
8  ARG1=T1*(I=1)
   FAC=(EXP(ARG1=1,0)*T3
9  CONTINUE
   X(I,J)*=(X(I,1)*(X(I,JMAX)=X(I,1))*FAC)*DFAC+(X(I,JMAX,J)=X(I,J)
1)*FAC*(Y(I,1))+Y(I,JMAX)=Y(I,1))*FAC)*DFAC+(Y(I,JMAX,J)=Y(I,J)
1)*FAC+Y(I,J)))*DFAC
   IF (XGE85,EXP.0,10,0,GE.S) GO TO 10
   IF (DFAC,EXP.0,0,0,DFAC,EXP.0,0) GO TO 10
   X(I,J)=0.5*X(I,J)
   Y(I,J)=0.5*Y(I,J)
10  CONTINUE
20  CONTINUE
C  RE-ENTER SEGMENT
C 30  IF (NRSEG,EXP.0) GO TO 140
   DO 130 L=1,NRSEG
      I1=LR1(L)
      I2=LR2(L)
      I1=I1+1
      I2=I2+1
      IGO=LRBID(L)
   GO TO (40,60,40,60), IGO
40  IF (LRBID(L),EQ.1) J=1
   IF (LRBID(L),EQ.3) J=JMAX
   DX=X(I2,J)=X(I1,J)
   DY=Y(I2,J)=Y(I1,J)
   DX=DX/(I2=I1)
   DY=DY/(I2=I1)
   DO 50 I=I11,12
      DX=X(I,1)=X(I,11)+DX
      DXY=I(I,11)=DY
      Y(I,J)=Y(I,J)+DY
   50  GO TO 80
60  IF (LRBID(L),EQ.2) J=1
   IF (LRBID(L),EQ.4) I=JMAX
   DX=X(I,12)=X(I,11)
   DXY=Y(I,12)=Y(I,11)
   DX=DX/(I2=I1)
   DY=DY/(I2=I1)
   DO 70 J=111,12
      DX=J=1)=DX
      DXY=J=1)=DY
      X(I,J)=X(I,1,1)+DX
   70  Y(I,J)=Y(I,1)+DY
60  I1=LI1(L)
   I2=LI2(L)
   I1=I1+1
   I2=I2+1
   IGO=LRBID(L)
   GO TO (90,110,90,110), IGO
90  IF (LRBID(L),EQ.1) J=1
   IF (LRBID(L),EQ.3) J=JMAX
   DX=X(I2,J)=X(I1,J)
   DXY=Y(I2,J)=Y(I1,J)
   DX=DX/(I2=I1)
   DXY=DY/(I2=I1)
DO 100 I=I1,II2
DDX(I-I1)=DX
DDY(I-I1)=DY
X(I,J)=X(I1,J)+DDX
Y(I,J)=Y(I1,J)+DDY
100 GO TO 130
110 IF (L1BD(L),EQ.2) I=1
IF (L1BD(L),EQ.4) I=IMAX
DX=X(I1,I2)-X(I1,I1)
DY=Y(I1,I2)-Y(I1,I1)
DX=DX/(I2-I1)
DY=DY/(I2-I1)
DO 120 J=I1,II2
DDX(J-I1)=DX
DDY(J-I1)=DY
X(I,J)=X(I1,J)+DDX
Y(I,J)=Y(I1,J)+DDY
120 CONTINUE
130 CONTINUE
140 IF (IGEB,EQ.0) RETURN
C
C LINEAR PROJECTION = IGES=1
C
IF (IGEB,NE,1) GO TO 170
DFAC=1.0
DFACC=1.0
IF(IGEB,EQ.1) DFACC=0.0
IF(IGEB,EQ.2) DFACC=0.0
DO 160 J=2,IMAX
FACC=FLOAT(J=1)/FLOAT(IMAX)
DO 150 I=2,IMAX
FACC=FLOAT(J=1)/FLOAT(IMAX)
X(I,J)=X(I1,J)+X(I,IMAX)-X(I1,1)*FAC)*DFAC+(X(IMAX,J)-X(1,J)
1)*FACC+X(1,J)*DFACC
Y(I,J)=Y(I1,J)+Y(I,IMAX)=Y(I1,1)*FAC)*DFAC+(Y(IMAX,J)-Y(1,J)
1)*FACC+Y(1,J)*DFACC
IF(DFAC,GT,0.0,OR,DFACC,GT,0.0) GO TO 150
X(I,J)=0.0,SX(I,J)
Y(I,J)=0.0,SY(I,J)
150 CONTINUE
160 CONTINUE
RETURN
C
C MOMENT OR EXPONENTIAL PROJECTION
C
170 SM=0.0,(IMAX+JMAX)=4
DM=SQRT(FLOAT((IMAX+1)**2+(JMAX+1)**2))
DM=1.088(IGEB)/DM
DO 210 J=2,IMAX
DO 210 I=2,IMAX
SX=X(I,J)+X(I,J)*X(II,I)+X(II,J)*MAX
SY=SY+Y(I,J)*Y(II,J)
210 CONTINUE
D=IABS(T1=1)
D1=IABS(1=J)
D2=IABS(JMAX=J)
D=D**2
D1=IABS(D+D1**2)
D2=IABS(D+D2**2)
SD=SD+D1+D2
SX=8XD+X(TI,1)+D1+X(TI,JMAX)*D2
SY=8YD+Y(TI,1)+D1+Y(TI,JMAX)*D2
E1=EXP(-D1*DM)
E2=EXP(-D2*DM)
SX=8XM*X(TI,1)+E1*X(TI,JMAX)*E2
SY=8YM*Y(TI,1)+E1*Y(TI,JMAX)*E2
SE=SE+E1+E2

180 CONTINUE

DO 190 JJ=2,JMAX+1
SX=8X*X(1,1)+X(IMAX,JJ)
SY=8Y*Y(1,1)+Y(IMAX,JJ)
D=IABS(JJ=J)
D1=IABS(1=1)
D2=IABS(IMAX=J)
D=D**2
D1=IABS(D+D1**2)
D2=IABS(D+D2**2)
SD=SD+D1+D2
SX=8XD+X(1,1)+D1+X(IMAX,JJ)*D2
SY=8YD+Y(1,1)+D1+Y(IMAX,JJ)*D2
E1=EXP(-D1*DM)
E2=EXP(-D2*DM)
SX=8XM*X(1,1)+E1*X(IMAX,JJ)*E2
SY=8YM*Y(1,1)+E1*Y(IMAX,JJ)*E2
SE=SE+E1+E2

190 CONTINUE

IF (IGEQ,EG,3) GO TO 200

C EXPONENTIAL PROJECTION = IGE8<0

T1=1.0/SE
X(I,J)=T1*8XM
Y(I,J)=T1*8YM

GO TO 210

200 CONTINUE

C MOMENT PROJECTION = IGE8=3 ON 4

T1=1.0/SD
IF (IGEQ,EG,3) T1=T1/BN
X(I,J)=(SD*B*X=BXD)*T1
Y(I,J)=(SD*B*Y=YD)*T1

210 CONTINUE

RETURN

C END

SUBROUTINE MAXMIN (X,IMAX,JMAX,NDIM,XMAX,XMIN,IMX,JMX,IMN,JMN,IBKI 1P1,IBKIP2)
C ******* THIS SUBROUTINE CALCULATES THE MAXIMUM AND MINIMUM VALUES *******
C ******* (IF A TWO-DIMENSIONAL DATA ARRAY.  *******
C ******
C  INPUT DATA:
C ******
C  X = 2-D DATA ARRAY WHOSE MAXIMUM & MINIMUM IS TO BE DETERMINED
C  IMAX = LARGEST VALUE OF FIRST SUBSCRIPT OF X TO BE SCANNED
C  JMAX = LARGEST VALUE OF SECOND SUBSCRIPT OF X TO BE SCANNED
C  NDIM = 1ST DIMENSION OF X
C  ISKIP1 = SKIP PARAMETER FOR 1ST INDEX OF X (THE I INDEX)
C  ISKIP2 = SKIP PARAMETER FOR 2ND INDEX OF X (THE J INDEX)
C ******
C  OUTPUT DATA:
C ******
C  XMAX = MAXIMUM OF X ARRAY
C  JXMAX = LOCATION OF XMAX
C  XMIN = MINIMUM OF X ARRAY
C  JXMIN = LOCATION OF XMIN
C **************************************************************
C  DIMENSION X(NDIM,1)
C ****************************************************************************
C  XMAX = X(I,J)
C  JXMAX = I,J
C  XMIN = X(I,J)
C  JXMIN = I,J
C  DO 20 I = 1,IMAX,ISKIP1
C  DO 20 J = 1,JMAX,ISKIP2
C  IF (X(I,J) .LT. XMAX) GO TO 10
C  XMAX = X(I,J)
C  JXMAX = I,J
C  10 IF (X(I,J) .GT. XMIN) GO TO 20
C  XMIN = X(I,J)
C  JXMIN = I,J
C  20 CONTINUE
C  RETURN
C  END
C
C SUBROUTINE PARA (XXI,YXI,XETA,YETA,LACC,CFAC,RETA,RXI,CSI,CSJ,WACC,1,R,T,ACCL,ACCL1,NDIM,1,J,IMER,TACC,IEV)
C ****************************
C  COEFFICIENTS AND VARIABLE ACCELERATION PARAMETERS ON
C  RE-ENTRANT SEGMENTS
C ****************************
C  DIMENSION RETA(NDIM,1), RXI(NDIM,1), WACC(NDIM,1), R(1), T(1)
C  DIMENSION TACC(NDIM,1)
C  DIMENSION IMER(1)
C  REAL JACBS,JSAG
C  INTEGER TACC
LOGICAL LACC

C****
UACC(RJ)=2.0/(1.0+SQRT(1.0+RJ**2))
GACC(RJ)=2.0/(1.0+SQRT(1.0+RJ**2))
ALFA*ETA**2+YETA**2
GAMA**XIXI**2+YXI**2
AG=1.0/(ALFA+GAMA)
JACC=(XIXI*YETA-XETA*YXI)*2*CFAC
J3AG=JACC*AG=0.0625
C
VARIABLE ACCELERATION PARAMETER

C
IF (LACC) GO TO 110
TEMP=JACC=0.125
B1=TEM1*RXI(I,J)
R2=TEM1*RETA(I,J)
B1=ABS(B1)
R2=ABS(R2)
T1=ALFA**2+B1**2
T2=GAMA**2+R2**2
AT1=ABS(T1)
AT2=ABS(T2)
IF (T1,GE,0.0,AND,T2,GE,0.0) GO TO 80
IF (T1,LT,0.0,AND,T2,LT,0.0) GO TO 90
RJ1=SQRT(AT1)*CSI*AG
IF (T1,GE,0.0) GO TO 10
TEMP=UACC(RJ1)
TACC(I,J)=10
GO TO 20
10 TEMP=UACC(RJ1)
TACC(I,J)=20
20 RJ2=SQRT(AT2)*CSI*AG
IF (T2,GE,0.0,AND,T2,LE,0.0) GO TO 30
TEMP=UACC(RJ2)
TACC(I,J)=TACC(I,J)+1
GO TO 40
30 TEMP=2*UACC(RJ2)
TACC(I,J)=TACC(I,J)+2
40 IF (IEV) 50,60,70
50 TEMP=UACC(SQRT(RJ1**2+RJ2**2))
GO TO 100
60 TEMP=(RJ1*TEMP1+RJ2*TEMP2)/(RJ1+RJ2)
GO TO 100
70 TEMP=UACC(SQRT(RJ1**2+RJ2**2))
GO TO 100
80 RJN(SQRT(AT1)*CSI+SQRT(AT2)*CSI)*AG
TEMP=UACC(RJ)
TACC(I,J)=1
GO TO 100
90 RJN(SQRT(AT1)*CSI+SQRT(AT2)*CSI)*AG
TEMP=UACC(RJ)
TACC(I,J)=1
100 R(11)=ERROR(TEMP*,WACC(I,J),R(11),I,J,IWER)
WACC(I,J)=TEMP
110 CONTINUE
C
COEFFICIENTS

C
T(5)=JSAG*RETA(I,J)
T(4)=JSAG*RXI(I,J)
SUBROUTINE RHS (IMAX, JMAX, NDIM, NLN, NPT, ATYP, ITYP, DEC, AMPFAC, ETA, I)

********* INHOMOGENEOUS TERM FOR COORDINATE ATTRACTION ***************

DIMENSION R(1), RETA(NDIM,1), ALN(20), DLN(20), JLN(20), APT(100),
1 DPT(100), IPT(100), JPT(100)
INTEGER XI, ETA, BLK, ATYP
DATA ZERO/1,N=1/
DATA XI, ETA, BLK/6XH, 6H

C****
IF (NLN,NE,O) RETURN
WRITE (6,200)
IF (ITYP,NE,O) WRITE (6,260)
IF (ITYP,N,NE,O) WRITE (6,270)
IF (ATYP,NE,ETA) WRITE (6,280)
IF (ATYP,NE,XI) WRITE (6,290)

C SET UP AMPLITUDE AND DECAY FACTOR
IF (NLN,NE,O) READ (5,210) (JLN(L),ALN(L),DLN(L),L=1,NLN)
IF (NPT,NE,O) READ (5,220) (IPT(L),JPT(L),APT(L),DPT(L),L=1,NPT)
IF (DEC,LT,ZERO) GO TO 40
IF (NLN,NE,O) GO TO 20
DO 10 L=1,NLN
10 DLN(L)=DEC
20 IF (NPT,NE,O) GO TO 40
DO 30 L=1,NPT
30 DPT(L)=DEC
40 CONTINUE
IF (ABS(AMPFAC),LT,ZERO) GO TO 90
IF (NLN,NE,O) GO TO 60
DO 50 L=1,NLN
50 ALN(L)=ALN(L)*AMPFAC
60 IF (NPT,NE,O) GO TO 90
DO 70 L=1,NPT
70 APT(L)=APT(L)*AMPFAC
80 CONTINUE
IF (NLN,NE,O) WRITE (6,230) (JLN(L),ALN(L),DLN(L),L=1,NLN)
IF (NPT,NE,O) WRITE (6,240) (IPT(L),JPT(L),APT(L),DPT(L),L=1,NPT)

C CALCULATE INHOMOGENEOUS TERM
DO 190 JM=1,IMAX
DO 190 JM=1,IMAX
C*** LINE ATTRACTION
IF (ATYP.EQ.ETA) IJ=J
IF (ATYP.EQ.XI) IJ=I
T2=0.0
IF (NLN.EQ.0) GO TO 100
DO 90 L=1,NLN
  T=ALN(L)*EXP(-DLN(L)*IHXS(IJ=JLN(I)))
  IF (ITYP.EQ.0) GO TO 90
  T=I8HN(1,1J=JLN(L))
  IF (IJ.EQ.JLN(L)) T=0.0
  T2=T2+T
90 CONTINUE
C*** POINT ATTRACTION
100 IF (*PT.EQ.0) GO TO 180
  DO 170 L=1,NPT
    I1=PT(L)
    IF (ATYP.EQ.ETA) GO TO 140
    I1=JPT(L)
    GO TO 150
  140 IF (LTNE) GO TO 170
  150 I1=IPT(L)
  160 IF (ITYP.EQ.0) 180
  170 CONTINUE
180 CONTINUE
C C C C C C C
C PRINT INHOMOGENEOUS TERM
C C C C C C C
C IF (IRIT.EQ.0) WRITE (6,250) ((I,J,RETA(I,J),T1,TM1),J=1,IMAX),J=1,JMAX)
C C C C C C C
C**** RETURN
C 200 FORMAT (*0EXPONENTIAL DECAY RHS*)
210 FORMAT (115,2F10.0)
220 FORMAT (215,2F10.0)
230 FORMAT (*0ATTRACTION LINES*://4X,*J*,17X,*AMP*,15X,*DECAY*/(15,2F20.1,8))
240 FORMAT (*0ATTRACTION POINTS*://4X,*I*,17X,*AMP*,15X,*DECAY*,
  1/(215,2F20.8))
250 FORMAT (*0RMS*://6(3X,*I*,3X,*J*,6X,*RETA **)/(6(2I4,1E10,4,* **))
260 FORMAT (* = ATTRACTION **)
270 FORMAT (* = ATTRACTION TO CONVEX SIDE, REPULSION TO CONCAVE **)
280 FORMAT (*0== ETA EQUATION RHS ====)
290 FORMAT (*0== XI EQUATION RHS ====)
C END
* LOGICAL CONTROLS: LACC = VARIABLE ACCELERATION PARAMETER FIELD
   CONVERGED, (INHIBIT IF LCON=TRUE)
   LCFAC = ADJUSTMENT OF INHOMOGENEOUS TERM COMPLETED
   LCON = CONSTANT ACCELERATION PARAMETER.
   ACC = ACCELERATION PARAMETER TYPE: 1 = BOTH EV IMAGINARY (UNDER=RELAX)
   2 = BOTH EV REAL (OVER=RELAX)
   12 = XI EV IMAGINARY, ETA EV REAL
   21 = XI EV REAL, ETA EV IMAGINARY

** DIMENSION X(NDIM,1), Y(NDIM,1), H(1), IXER(1), IYER(1), T(5), LR1(11), LR2(1), L11(11), MT2(1), LTYPE(1), IHER(1)
** DIMENSION =ACC(NDIM,1), RETA(NDIM,1), XI(NDIM,1), TACC(NDIM,1)
** DOUBLE PRECISION AG
** INTEGER ATYP, ETA, XI, TACC
** REAL JSAG, JACRS
** LOGICAL LACC, LCFAC, LCON
** DATA XI, ETA /6HIXI, ETA/
** DATA PI/3.14159265359/

C***
** HACK(X,T,I,J,I1,I2,J1,J2,A,A1,T2,T1)MA1=X(I,J)*A*T(1)*(X(I2,1)+X(11,1))
** 1*(X(I2,1)+X(11,1)-X(I1,1)*T(1)*T(1))
** 1*(X(I,J2)+X(1,J1))=T(3)*(X(I2,J2)-X(I2,J1)+X(I1,J1)=X(I1,J2))
** 2*(X(I1,J2))=T(5)*T2*T(4)*T(4)*T(1)
** HACK2(X,T,I,J,I1,I2,J1,J2,A,A1,T2,T1)MA1=X(I,J)*A*T(1)*(X(I2,J)+X(11,1))
** 1*(X(I,J2)+X(1,J1))=T(1)*(X(I2,J2)+X(I1,J1))=X(I1,J2)+X(I2,J2)
** 2*(X(I1,J2)+X(1,J1))=T(3)*(X(I1,J1)+X(I2,J2)+X(I1,J2))
** 2*(X(I1,J1))=T(5)*T2*T(4)*T(4)*T(1)
** HACK3(X,T,I,J,I1,I2,J1,J2,A,A1,T2,T1)MA1=X(I,J)*A*T(1)*(X(I2,J)+X(11,1))
** 1*(X(I,J2)+X(1,J1))=T(1)*(X(I2,J2)+X(I1,J1)+X(I2,J1))
** 2*(X(I2,J1))=T(3)*(X(I2,J1)+X(I2,J2)+X(I1,J2))
** 2*(X(I2,J2))=T(5)*T2*T(4)*T(4)*T(1)
** HACK5(X,T,I,J,I1,I2,J1,J2,A,A1,T2,T1)MA1=X(I,J)*A*T(1)*(X(I2,J)+X(11,1))
** 1*(X(I,J2)+X(1,J1))=T(1)*(X(I2,J2)+X(I1,J1)+X(I2,J1))
** 2*(X(I2,J1))=T(3)*(X(I2,J1)+X(I2,J2)+X(I1,J2))
** 2*(X(I2,J2))=T(5)*T2*T(4)*T(4)*T(1)
** UACC(RJ)R2/1.0-SQRT(1.0-RJ**2)
** OACC(RJ)R2/1.0+SQRT(1.0-RJ**2)

C***
** WRITE (6,630)
C
** READ COORDINATE ATTRACTION PARAMETERS AND CALCULATE
** INHOMOGENEOUS TERM
C
** KO=1
** READ (5,610) ATYP, ITYP, NLN, NPT, DEC, AMPFAC
** IF (ATYP.EQ.ETA) CALL RHB (R, IMAX, JMAX, NDIM, NLN, NPT, ATYP, ITYP, DEC, AMPFAC, RETA)
** IF (ATYP.EQ.XI) CALL RHB (R, IMAX, JMAX, NDIM, NLN, NPT, ATYP, ITYP, DEC, AMPFAC, RXI)
** READ (5,610) ATYP, ITYP, NLN, NPT, DEC, AMPFAC
** IF (ATYP.EQ.ETA) CALL RHB (R, IMAX, JMAX, NDIM, NLN, NPT, ATYP, ITYP, DEC, AMPFAC, RETA)
** IF (ATYP.EQ.XI) CALL RHB (R, IMAX, JMAX, NDIM, NLN, NPT, ATYP, ITYP, DEC, AMPFAC, RXI)
IMPFAC,RX1)
READ (5,590) IFAC,IRIT,EFAC
C INITIAL SETUP
C IF (R(1),GT,0.0) GO TO 40
  IF (IEV) 10,20,30
10 WRITE (6,640)
    GO TO 40
20 WRITE (6,650)
    GO TO 40
30 WRITE (6,660)
40 CONTINUE
  IF (IDISK,EQ,2,OR,IDISK,EQ,3) GO TO 50
  GO TO 60
50 READ (11,720) LACC,LCON,CFAC,LCFAC,INCOUN,INCOUN,CIFAC,IEV,IFAC
  1,EFAC,JMXM1,IMXM1,JMXP1,C81,CSJ,NPT,INFAC,INFIN,END
READ (11,730) (R(I),I=1,13)
READ (11,750) (X(I,J),Y(I,J),RXI(I,J),RETA(I,J),WACC(I,J),?M1,IMA
1X),JM1,JMAX)
REIND 11
  IF (IEND,NE,3) GO TO 100
  K01=1
  GO TO 80
60 IF (IFAC,EQ,0) IFAC=1
  WRITE (6,600) IFAC,EFAC
  EFAC=MIN1(R(2),R(3))*EFAC
  IF (IFAC,GT,0) CFAC=1.0/FLOAT(2**IFAC-1)
  IF (IFAC,LT,0) CFAC=1.0/ABS(FLOAT(IFAC))
  WRITE (6,670) CFAC
  INCOUN1=1
  IF (IWR,GT,0) WRITE (6,570)
    JM1=1,MAX1=1
    IMXM1=MAX1=1
    JMXP1=JMAX1=1
    CSJ=COS(PI/FLOAT(IMXM1))
    CS1=COS(PI/FLOAT(JMXP1))
    NPT=JMAX1=2)
    IF (NRSEG,GT,0) GO TO 75
    DO 10 L=1,NRSEG
70  NPT=NPT+L2(L)=LR1(L)=1
75 IF (INFAC,EQ,0) INFAC=1
  IF (INFAC,GT,0) CINFAC=1.0/FLOAT(2**INFAC-1)
  IF (INFAC,LT,0) CINFAC=FLOAT(INFAC)/ABS(FLOAT(INFAC))
  WRITE (6,710) CINFAC
  INCOUN1=1
  IF (INFAC,LT,0) INCOUN=INFAC
  ENINF=EFAC
C SET UP ITERATION
80 LCON=NS,GT,0.0
  LCFAC=FALSE
  IF (IABS(IFAC),LE,1) LCFAC=TRUE
  DO 90 IM1,JMAX
    DO 90 J=1,JMAX
90  WACC(I,J)=1.0
    WRITE (6,620)
SOR ITERATION

100 IEND=0
DO 450 K=0,ITER
   IF (K.GE.2) LACC=.TRUE.,
   IF (K.GE.3) GO TO 120
   LACC=.FALSE.,
   IF (LCON) LACC=.TRUE.,
   TM=R(1)
   DO 110 I=1,I MAX
      ACC(I,J)=TM
   DO 110 J=1,J MAX

110 R(4)=0.0
   R(5)=0.0
   R(11)=0.0
   R(12)=0.0

C**** FIELD ****

DO 260 JJ=2,J MAX1
   J=J MAX1=J+2
   JM=J+1
   JP=J+1
   DO 260 II=2,IMAX1
      IMAX1=II+2
      IP1=II+1
      IF (I,GT,1) GO TO 130
      IM=IMAX1
      GO TO 140
   I=I+1
140
   XXI=X(IP1,J)=X(IM1,J)
   YYI=Y(IP1,J)=Y(IM1,J)
   XETA=X(I,JP1)=X(I,JM1)
   YETA=Y(I,JP1)=Y(I,JM1)
   ALFA=XETA**2+YETA**2
   GAMMA=XXI**2+YYI**2
   AG=1.0/ALFA*GAMA
   JACBB=XXI*YETA-XETA*YYI**2*CFAC
   JACBB=JACBB*AG*0.0625

C**** VARIABLE ACCELERATION PARAMETER

   IF (LACC) GO TO 250
   TEM1=JACBB*0.125
   B1=TEM1*XI(I,J)
   B2=TEM1*ETA(I,J)
   B1=ABS(B1)
   B2=ABS(B2)
   T1=ALFA**2*B1**2
   T2=GAMA**2*B2**2
   AT1=ABS(T1)
   AT2=ABS(T2)
   IF (T1.GE.0.0,AND.T2.GE.0.0) GO TO 220
   IF (T1.LT.0.0,AND.T2.LT.0.0) GO TO 230
   RJ1=SGRT(AT1)*CB1*AG
   IF (T1.GE.0.0) GO TO 150
   TEM1=UACC(RJ1)
   TACC(I,J)=10
   GO TO 160
150 TEM1=OACC(RJ1)
   TACC(I,J)=20
160 RJ1=SGRT(AT2)*CBJ*AG
IF (T2,GE,0.0) GO TO 170
TEMP=2*OACC(RJ)
TACC(I,J)=TACC(I,J)+1
GO TO 180
170
TEMP=2*OACC(RJ)
TACC(I,J)=TACC(I,J)+2
180
IF (IEV) 190,200,210
190
TEMP=UACC(SORT(RJ1**2+RJ2**2))
GO TO 200
200
TEMP=(RJ1*TEMP1+RJ2*TEMP2)/(RJ1+RJ2)
GO TO 240
210
TACC(I,J)=2
GO TO 240
220
RJm(SORT(AT1)*CS1+SORT(AT2)*CSJ)*AG
TACC(I,J)=TACC(RJ)
GO TO 240
230
RJm(SORT(AT1)*CS1+SORT(AT2)*CSJ)*AG
TACC(I,J)=TACC(RJ)
GO TO 240
240
R(11)*ERROR(TEMPm+ACC(I,J),R(11),I,J,IWER)
WACC(I,J)=TEMP
CONTINUE
C**** ITERATE
T(5)=JBag*Eta(I,J)
T(4)=JBag*Rxi(I,J)
T(1)=5*Alfa*AG
T(2)=5*Gama*AG
T(3)=25*(X*Eta+Y*Yxi*Eta)*AG
ACCL=WACC(I,J)
ACCL1=ACCL+ACCL
R(12)=R(12)+ACCL
TEMP=WACK(X,T,I,J,JM1,JM1,JM1,JM1,ACCL,ACCL1,XFGA,XG)
R(4) ERROR(X,T,I,J,JM1,JM1,JM1,JM1,ACCL,ACCL1,XFGA,XG)
R(5) ERROR(Y,T,I,J,JM1,JM1,JM1,JM1,ACCL,ACCL1,YFGA,YG)
X(I,J)=TEMP
Y(I,J)=TEMP
IF (I,J,GT,1) GO TO 260
X(IMAX,J)=TEMP
Y(IMAX,J)=TEMP
CONTINUE
C
C**** REENTRANT SEGMENTS ***
C
IF (NRSEG.EQ,0) GO TO 400
DO 390 L=1,NRSEG
N1=L+1
N2=L+1
N3=L+1
N4=L+1
IGOTOLTYPE(L)
GO TO (270,290,310,330,350,370), IGOTO
C**** ONE ON BOTTOM, ONE ON TOP
270 DO 280 JJ=11,12
I2=JJ+1
XXIMX(I+1,1)=X(I=1,1)
YIMY(I+1,1)=Y(I=1,1)
XETAMX(I,2)=X(I,JMXM)
YETAMY(I,2)=Y(I,JMXM)
CONTINUE
CALL PARA (XXI, XSI, XETA, YETA, LACC, CFAC, RETA, RXI, CSI, CSJ, MACC)
1, R, T, ACCL, ACCL1, NDIM, I, IFR, TACC, IEV)

CONTINUE
GO TO 390
C**** HOME ON BOTTOM
390 DO 300 J = I1, I2
  I = I + 1
  IT = IT + (T = I)
  XXI = XXI + (X(I = I + 1))
  YXI = YXI + (Y(I = I + 1))
  XETA = XETA + (X(I = I + 1))
  YETA = YETA + (Y(I = I + 1))
  CALL PARA (XXI, XSI, XETA, YETA, LACC, CFAC, RETA, RXI, CSI, CSJ, MACC)
1, R, T, ACCL, ACCL1, NDIM, I, IFR, TACC, IEV)

CONTINUE
GO TO 390
C**** HOME ON TOP
310 DO 320 J = I1, I2
  I = I + 1
  IT = IT + (T = I)
  XXI = XXI + (X(I = I + 1))
  YXI = YXI + (Y(I = I + 1))
  XETA = XETA + (X(I = I + 1))
  YETA = YETA + (Y(I = I + 1))
  CALL PARA (XXI, XSI, XETA, YETA, LACC, CFAC, RETA, RXI, CSI, CSJ, MACC)
1, R, T, ACCL, ACCL1, NDIM, I, IFR, TACC, IEV)

CONTINUE
GO TO 390
C**** ONE ON LEFT, ONE ON RIGHT
330 DO 340 J = I1, I2
  J = J + 1
  XXI = XXI + (X(I + J = I) + X(I + J = J))
  YXI = YXI + (Y(I + J = I) + Y(I + J = J))
  XETA = XETA + (X(I + J = I) + X(I + J = J))
  YETA = YETA + (Y(I + J = I) + Y(I + J = J))
  CALL PARA (XXI, XSI, XETA, YETA, LACC, CFAC, RETA, RXI, CSI, CSJ, MACC)
1, R, T, ACCL, ACCL1, NDIM, J, JER, TACC, IEV)
   TEMPX = MACK(X, T, J, IMX1, J, 1, J, 1, ACCL, ACCL1, XETA, XXI)
   TEMPY = MACK(Y, T, J, IMX1, J, 1, J, 1, ACCL, ACCL1, YETA, XXI)
   R(4) = ERROR(TEMPX, X(1, J), R(4), I, J, IYER)
   R(5) = ERROR(TEMPY, Y(1, J), R(5), I, J, IYER)
   X(1, J) = TEMPX
   Y(1, J) = TEMPY
   X(IMAX, J) = TEMPX
   Y(IMAX, J) = TEMPY
340 CONTINUE
GO TO 340
C++++ ROTH ON LEFT
350 DO 360 JJ = I1, I2
   J = J + 1
   II = I4(J + 1)
   XXI = X(IMX1 + J, II)
   X = Y(IMX1 + J, II)
   XETA = X(IMAX + J + 1, XXI)
   YETA = Y(IMAX + J + 1, XXI)
   CALL PARA (XXI, X, XETA, YETA, LACC, CFAC, RETA, RXI, CSJ, CB, HACC
1, R, T, ACCL, ACCL1, NDIM, J, JER, TACC, IEV)
   TEMPX = MACK(X, T, J, IMX1, J, 1, J, 1, ACCL, ACCL1, XETA, XXI)
   TEMPY = MACK(Y, T, J, IMX1, J, 1, J, 1, ACCL, ACCL1, YETA, XXI)
   R(4) = ERROR(TEMPX, X(1, J), R(4), I, J, IYER)
   R(5) = ERROR(TEMPY, Y(1, J), R(5), I, J, IYER)
   X(1, J) = TEMPX
   Y(1, J) = TEMPY
   X(1, II) = TEMPX
   Y(1, II) = TEMPY
360 CONTINUE
GO TO 370
C++++ ROTH ON RIGHT
370 DO 380 JJ = I1, I2
   J = I + 1
   II = I4(II + 1)
   XXI = X(IMX1, II) - X(IMX1, J)
   X = Y(IMX1, II) - Y(IMX1, J)
   XETA = X(IMAX, J + 1, XXI)
   YETA = Y(IMAX, J + 1, XXI)
   CALL PARA (XXI, X, XETA, YETA, LACC, CFAC, RETA, RXI, CSJ, CB, HACC
1, R, T, ACCL, ACCL1, NDIM, J, JER, TACC, IEV)
   TEMPX = MACK(X, T, J, IMX1, J, 1, J, 1, ACCL, ACCL1, XETA, XXI)
   TEMPY = MACK(Y, T, J, IMX1, J, 1, J, 1, ACCL, ACCL1, YETA, XXI)
   R(4) = ERROR(TEMPX, X(IMAX, J), R(4), IMAX, J, IYER)
   R(5) = ERROR(TEMPY, Y(IMAX, J), R(5), IMAX, J, IYER)
   X(IMAX, J) = TEMPX
   Y(IMAX, J) = TEMPY
   X(IMAX, II) = TEMPX
   Y(IMAX, II) = TEMPY
380 CONTINUE
390 CONTINUE
400 CONTINUE
C
C $STORE INITIAL ITERATION ERROR NORM8
C
C R(12) = (12) / FLOAT(NPT)
IF (K, GT, 1) GO TO 410
R(6) = R(4)
R(7) = R(5)
C WRITE ITERATION ERROR NORM

C 410 IF (IWR.GT.0) WRITE (6,540) K,R(4),R(5),R(11),R(12),LACC,LCFAC,
     1 ILCON
C CHECK TO SEE IF ITERATION IS COMPLETE
C
C**** CHECK VARIABLE ACCELERATION PARAMETER FIELD CONVERGENCE
   LACC[R(11)].LT.R(10)
   IF (ILCON) LACC.TRUE.
   IF (LACC) R(11)=0.0
C**** CHECK INTERMEDIATE FIELD CONVERGENCE BEFORE INCREASING
C**** COORDINATE ATTRACTION
   IF (LCFAC) GO TO 420
   IF (R(4),GT.0,EVAC,OR,R(5),GT.EVAC) GO TO 420
C**** INCREASE COORDINATE ATTRACTION
   R(4)=1.0
   LACC.,FALSE.
   IF (ILCON) LACC.,TRUE.
   IF (IFAC),GT.0 CFAC=2.0*CFAC
   IF (IFAC).LT.0 CFAC=CFAC*FLOAT(IROCOUN+1)/FLOAT(INCOUN)
   IRCON=IRCON+1
   IF (CFAC.GT.1.0) CFAC=1.0
   LCFAC=ABS(CFAC=1.0),LT,0.000001
   WRITE (6,670) CFAC
   420 CONTINUE
C**** CHECK INTERMEDIATE FIELD CONVERGENCE BEFORE MOVING
C**** OUTER BOUNDARY
   IF (INCON.GE.IABS(INFAC)) GO TO 430
   IF (R(4),LT.EVAC,AND,R(5),LT.EVAC) GO TO 490
C**** CHECK FIELD CONVERGENCE
   430 IF (R(4),GT.R(2),AND,R(5),LT.R(3)) GO TO 490
C**** PRINT VARIABLE ACCELERATION PARAMETER FIELD AT START OF ITERATION
   IF (IAIT.EQ.0) GO TO 450
   WRITE (6,680)
   DO 440 J=1,JMAX
       WRITE (6,690) J
   ENDIT
   WRITE (6,700) (TACC(I,J) WACC(I,J),IMAX)
   440 CONTINUE
   450 CONTINUE
C ITERATION DOES NOT CONVERGE
C
C**** WRITE PARTIALLY CONVERGED SOLUTION TO DISK
   455 WRITE (11,720) LACC,LCON,CFAC,LCFAC,IRCON,INCON,CINFAC,K,IEV,IFA
   1C,EFAC,JMXM1,IMXM1,IXM1,JKP1,CSI,CSI,NPT,INFAC,EINF,EEND
   WRITE (11,730) (R(I),I=1,13)
   WRITE (11,730) ((X(I,J),Y(I,J),RXI(I,J),RETA(I,J),WACC(I,J),IMAX)
   1IAX),J=1,jmax)
C**** PRINT VARIABLE ACCELERATION PARAMETER FIELD
   IF (IAIT.EQ.0) GO TO 470
   WRITE (6,680)
   DO 460 J=1,JMAX
       WRITE (6,690) J
   ENDIT
   WRITE (6,700) (TACC(I,J) WACC(I,J),IMAX)
   460 CONTINUE
   470 CONTINUE
   IF (R(6),GT.R(4),AND,R(7),GT.R(5)) GO TO 480
   IEND=1
RETURN
490 IEND$3
RETURN
C
C ITERATION CONVERGES
C
490 IEND$3.
C**** MOVE OUTER BOUNDARY
   DINFAC=CINFAC
   IF (INCOUN.GE.14R8C(INFAC)) GO TO 550
   IF (INFAC.GT.0) CINFAC=2.0*CINFAC
   IF (INFAC.LT.0) CINFAC=CINFAC*FLOAT(INCOUN+1)/FLOAT(INCOUN)
   DINFAC=CINFAC/OINFAC
   WRITE (6,710) CINFAC
   OINFAC=OINFAC
   INCOUN=INCON+1
   DO 540 LI$=ISEG,MBSEG
      I=LB1(L)
      I2=LB2(L)
      IGOTO=LAB3(L)
      GO TO (500,520,500,520), IGOTO
500   IF (IGOTO,EQ,1) J=1
      IF (IGOTO,EQ,3) J=JMAX
      DO 510 I=1,12
         X(I,J)*X(I,J)*DINFAC
510   Y(I,J)*Y(I,J)*DINFAC
      GO TO 540
520   IF (IGOTO,EQ,2) I=1
      IF (IGOTO,EQ,4) I=I+MAX
      DO 530 J=1,12
         X(I,J)*X(I,J)*DINFAC
530   Y(I,J)*Y(I,J)*DINFAC
      CONTINUE
      IF(LCFAC) IFAC=1
      K0=1
      IF(X,EQ,ITER) GO TO 455
      GO TO 80
C**** FINAL CONVERGENCE
550 CONTINUE
   ITER=K
C**** PRINT VARIABLE ACCELERATION PARAMETER FIELD
   IF (IAIT,EQ,0) RETURN
   WRITE (6,880)
   DO 560 J=1,JMAX
      WRITE (6,490) j
   WRITE (6,700) (TACC(I,J),WACC(I,J),I=1,JMAX)
560 CONTINUE
RETURN
C
570 FORMAT (1X,15X,*ITERATION ERROR NORMS*/)
580 FORMAT (1$3E15.5,F20.5,F10.0)
590 FORMAT (215,2F10.0)
600 FORMAT (40$----- MAXIMUM NORMS OF ITERATE CHANGES,40$,*LOGICAL CON-  
   TROL,40$,*ITERATE*,9X,*=NORM*,9X,*=NORM*,7X,*ACC=NORM*,7X,*AVG,  
   A 2CC,PARA.*,6X,*LACC,5X,*LCINF*,6X,*LCON*=)
610 FORMAT (40$----- MAXIMUM NORMS OF ITERATE CHANGES,40$,*LOGICAL CON-  
   TROL,40$,*ITERATE*,9X,*=NORM*,9X,*=NORM*,7X,*ACC=NORM*,7X,*AVG,  
   A 2CC,PARA.*,6X,*LACC,5X,*LCINF*,6X,*LCON*=)
620 FORMAT (40$----- MAXIMUM NORMS OF ITERATE CHANGES,40$,*LOGICAL CON-  
   TROL,40$,*ITERATE*,9X,*=NORM*,9X,*=NORM*,7X,*ACC=NORM*,7X,*AVG,  
   A 2CC,PARA.*,6X,*LACC,5X,*LCINF*,6X,*LCON*=)
SOLUTION PARAMETERS //20X

640 FORMAT (*COMPLEX EIGENVALUE PROCEDURE: UNDERRELAX*)
650 FORMAT (*COMPLEX EIGENVALUE PROCEDURE: HEIGHTED AVERAGE*)
660 FORMAT (*COMPLEX EIGENVALUE PROCEDURE: OVERRELAX*)
670 FORMAT (*.F15.8, IF INHOMOGENEOUS TEMPERATURE*)
680 FORMAT (*.ACCELERATION PARAMETERS//)
690 FORMAT (*.J =.I3/I)
700 FORMAT (10(I*4,(*12I,*,F7.4,1X)))
710 FORMAT (*.F15.8, OF OUTER BOUNDARY*)
720 FORMAT (2L10,E16.8,L10,215,E16.8,3I5,E16.8,3I5,2E1A,8/2I5,E16.8,15)
730 FORMAT (8E1A,8)

C
END

SUBROUTINE PLOT(X,Y,I)
CALL CALPLT(X,Y,I)
RETURN
END

SUBROUTINE SYMBOL(X,Y,M,TFX,T,Y,ANGLE,NCHAP)
CALL NOTATE(X,Y,M,TFX,T,Y,ANGLE,NCHAP)
RETURN
END

2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
Sample Cases
Sample Case Output: Simply-Connected Region

**************************** INPUT *****************************

MAXJ,MAXI,MINX,MAXX,MINY,MAXY,INITL,INITU,INSTC,ISUN,ISEED 1 21 21 0 100 0 1 1 0 0 0
ICPLOT,IPLOT,ICOPY,ILIN=1,ILINE=100,NMAXM,NUMS1,ISKIP1,ISKIP2 1 3 1 0 0 0 0 1 1
NBSEG,NUMSEG 1 0
H(1),H(2),H(3),HINF,XMIN,XINF,YMIN,YINF,DMIN 1,0000000 0,0000000 0,0000000 0,0000000 0,0000000 0,0000000 0,0000000 0,0000000 0
IEV,IAIT,H(10) 1 0 0 0,0000000
INFA1,INFAU 1 0

SIMPLY-CONNECTED REGION

INITIAL GUESS TYPE 0

--BODY SEGMENTS--

L 1 1 1 d1 =1
2 4 1 d1 =1
3 3 d1 1 =1
4 2 d1 1 =1

--OUTER BOUNDARY--

MAXIOS 0,0000000
INITIAL ANGLE 0,0000000
NORMAL AT X = 0,0000000 , Y = 0,0000000

NUMBER OF POINTS 0
1 STEPS IN ATTAINMENT OF INFINITY

INITIAL STEP (LINEAR CASE) 1

UNIFORM ACCELERATION PARAMETER USED.

TEST CASE = BODY-FITTED COORDINATE SYSTEM
SIMPLY-CONNECTED REGION

BODY-FITTED COORDINATE SYSTEM

TRANSFORMED BODY, KEY SEAT SHAFT

FIELD PARAMETERS, NUMBER OF XI-LINES 21
NUMBER OF ETA-LINES 21

ITERATION PARAMETERS: SUM ACCELERATION PARAMETER 1,000000
MAXIMUM NUMBER OF ITERATIONS ALLOWED 100
ALLOWABLE ITERATION ERROR NUMBER:.1,1000E-03
.1,1000E-03

NUMBER OF BODIES IN FIELD 1

PLOT PARAMETERS: NUMCORIES DESIRED 1
LINE HEIGHT DESIRED 0
PLOT SIZE IN THEMOTION 0,000
MATH 0,000

LOGICAL CONTROLS: LACC = VARIABLE ACCELERATION PARAMETER FIELD CONVERGED (IN RELEVANT IF LCONST TRUE)
LC0F = AQUATION OF INHOMOGENEOUS TERM COMPLETED
LC0N = UNIFORM ACCELERATION PARAMETER
Sample Cases
Sample Case Input: Simply-Connected Region

TEST CASE = BODY-FITTED COORDINATE SYSTEM
SIMPLY-CONNECTED REGION

KEY SEAT SHAFT

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>21</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>21</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>21</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.8 0.0001 0.0001 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0 0 0 0 0 0 0 0 0 0

8.0 0 0 0 10.0 0 0 0 0 0 0 0 0 0 0 0

= .125 .37461 = .62932 = .82904 = .9563 = .9563 = .9563 = .9563 = .9563 = .9563
= .37461 = .62932 = .82904 = .9563 = .125 = .125 = .125 = .125 = .125 = .125
= .37461 = .9563 = .9563 = .9563 = .9563 = .9563 = .9563 = .9563 = .9563 = .9563
= .125 = .125 = .125 = .125 = .125 = .125 = .125 = .125 = .125 = .125
= .9215 = .92718 = .92718 = .92718 = .92718 = .92718 = .92718 = .92718 = .92718 = .92718
= .9563 = .929237 = .929237 = .929237 = .929237 = .929237 = .929237 = .929237 = .929237 = .929237
= .986 = .978 = .978 = .962 = .954 = .946 = .936 = .936 = .932 = .932
= .914 = .908 = .908 = .908 = .908 = .908 = .908 = .908 = .908 = .908
<table>
<thead>
<tr>
<th>ETA</th>
<th>X</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.01</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>XI</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Sample Case Output: Simply-Connected Region

************** INPUT **************

INITIAL STEP (LINEAR CASE) = 1

TEST CASE = BODY=FITTED COORDINATE SYSTEM
SIMPLY-CONNECTED REGION

BODY=FITTED COORDINATE SYSTEM

TRANSLATED BODY, KEY SEAT SHAFT

FIELD PARAMETERS, NUMBER OF XI-LINES = 21
NUMBER OF ETA-LINES = 21

ITERATION PARAMETERS: SUM ACCELERATION PARAMETER = 1.000000
MAXIMUM NUMBER OF ITERATIONS ALLOWED = 100
ALLOWABLE ITERATION ERROR NUMBERS: X = 1.000E-03
Y = 1.000E-03

NUMBER OF BODIES IN FIELD = 1

PLOT PARAMETERS: COPIES DESIRED = 1
LINE LIGHT DESIRED = 0
PLOT SIZE IN Y-DIRECTION = 0.000
MATI = 0.000

LOGICAL CONTROLS: I LACC = VARIABLE ACCELERATION PARAMETER FIELD CONVERGED (IMRELEVANT IF LCON=TRUE)
LIFAC = AUDITION OF INHOMOGENEOUS TERM COMPLETED
LCON = UNIFORM ACCELERATION PARAMETER
**EXPONENTIAL DECAY RHS**

- **ATTRACTION**

--- ETA EQUATION RHS ---

**ATTRACTION LINES**

<table>
<thead>
<tr>
<th>J</th>
<th>AMP</th>
<th>DECAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0E6</td>
<td>0.001E6</td>
</tr>
</tbody>
</table>

1 steps in addition of inhomogeneous term, intermediate convergence factor = 100.00000000

1.0000000000 of inhomogeneous term

1.0000000000 of outer boundary

----- Maximum norms of iterate changes

<table>
<thead>
<tr>
<th>ITERATE</th>
<th>X-NORM</th>
<th>Y-NORM</th>
<th>ACC-NORM</th>
<th>AVG,ACC,PARA.</th>
<th>LACC</th>
<th>LCFAC</th>
<th>LEQN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.77E+01</td>
<td>1.37E+00</td>
<td>0.000000</td>
<td>1.000000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>7.35E-01</td>
<td>1.79E-00</td>
<td>0.000000</td>
<td>1.000000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>2.58E+00</td>
<td>5.24E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>1.92E+00</td>
<td>4.20E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>1.57E+00</td>
<td>3.67E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>6</td>
<td>1.93E+00</td>
<td>2.77E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>7</td>
<td>8.14E+00</td>
<td>2.39E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>8</td>
<td>2.08E+00</td>
<td>1.88E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>9</td>
<td>8.74E+00</td>
<td>1.40E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>10</td>
<td>4.75E+00</td>
<td>9.60E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>11</td>
<td>1.08E+00</td>
<td>6.90E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>12</td>
<td>2.99E+00</td>
<td>5.58E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>13</td>
<td>2.70E+00</td>
<td>5.77E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>14</td>
<td>1.76E+00</td>
<td>6.16E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>15</td>
<td>1.94E+00</td>
<td>4.52E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>16</td>
<td>1.90E+00</td>
<td>2.95E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>17</td>
<td>1.70E+00</td>
<td>1.37E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>18</td>
<td>1.40E+00</td>
<td>6.02E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>19</td>
<td>6.94E+00</td>
<td>3.31E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>20</td>
<td>9.90E+00</td>
<td>2.37E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>21</td>
<td>1.04E+00</td>
<td>2.29E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>22</td>
<td>1.11E+00</td>
<td>2.30E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>23</td>
<td>2.91E+00</td>
<td>1.13E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>24</td>
<td>7.10E+00</td>
<td>1.29E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>25</td>
<td>5.06E+00</td>
<td>1.37E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>26</td>
<td>6.85E+00</td>
<td>6.95E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>27</td>
<td>3.35E+00</td>
<td>6.43E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>28</td>
<td>2.93E+00</td>
<td>4.45E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>29</td>
<td>2.19E+00</td>
<td>3.28E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>30</td>
<td>1.90E+00</td>
<td>2.53E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>31</td>
<td>1.40E+00</td>
<td>2.79E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>32</td>
<td>1.02E+00</td>
<td>1.35E+00</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>33</td>
<td>7.66E-03</td>
<td>6.93E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>34</td>
<td>6.29E-03</td>
<td>6.36E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>35</td>
<td>4.90E-03</td>
<td>5.57E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>36</td>
<td>4.75E-03</td>
<td>6.25E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>37</td>
<td>2.85E-03</td>
<td>3.81E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>38</td>
<td>3.08E-03</td>
<td>3.02E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>39</td>
<td>3.28E-03</td>
<td>3.07E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>40</td>
<td>2.70E-03</td>
<td>1.13E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>41</td>
<td>1.64E-03</td>
<td>2.89E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>42</td>
<td>1.40E-03</td>
<td>2.89E-03</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>43</td>
<td>9.68E-04</td>
<td>1.98E-04</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>44</td>
<td>1.10E-04</td>
<td>1.42E-04</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>45</td>
<td>1.34E-04</td>
<td>1.45E-04</td>
<td>0.000000</td>
<td>1.800000</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

LOGICAL CONTROLS
TEST CASE = BUMPFITTED COORDINATE SYSTEM
SIMPLY CONNECTED REGION

FINAL VALUE

ITERATION CONVERGED.

INITIAL ITERATION ERROR NUMBER: XI .771760E+01 YI .78730E+00 AT ITERATE # 1
FINAL ITERATION ERROR NUMBER: XI .59074E+04 YI .85360E+00 AT ITERATE # 4b

LOCATION OF MAXIMUM ITERATION ERROR: XI = 5 YI = 17

<table>
<thead>
<tr>
<th>X-ARRAY</th>
<th>Y-ARRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1250E+04, .37461E+00, .2932E+00, .8290E+00, .6580E+00, .10000E+01, .59530E+00, .6290E+00, .4290E+00, .2932E+00, .37461E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .2815E+00, .4851E+00, .6366E+00, .7885E+00, .9359E+00, .1228E+00, .7254E+00, .5913E+00, .3128E+00, .2815E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .3128E+00, .3513E+00, .7254E+00, .8228E+00, .8359E+00, .7685E+00, .9359E+00, .4651E+00, .2815E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .4219E+00, .3811E+00, .6845E+00, .5933E+00, .6730E+00, .6747E+00, .4992E+00, .2906E+00</td>
<td></td>
</tr>
<tr>
<td>.11776E+04, .2569E+00, .4589E+00, .6977E+00, .4651E+00, .3811E+00, .2815E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .2162E+00, .2990E+00, .3612E+00, .4356E+00, .5033E+00, .5215E+00, .4786E+00, .3710E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .2932E+00, .4589E+00, .5056E+00, .5082E+00, .4556E+00, .3811E+00, .2990E+00, .2162E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1993E+00, .2607E+00, .3632E+00, .3917E+00, .3464E+00, .3591E+00, .2781E+00, .1532E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1532E+00, .2781E+00, .3943E+00, .3917E+00, .3620E+00, .3139E+00, .1807E+00, .1493E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1870E+00, .2357E+00, .2732E+00, .3015E+00, .3147E+00, .3047E+00, .2057E+00, .1116E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1116E+00, .2057E+00, .2732E+00, .3015E+00, .3147E+00, .3047E+00, .1807E+00, .1493E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1774E+00, .2150E+00, .2429E+00, .2622E+00, .2839E+00, .2501E+00, .1609E+00, .0854E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .6223E+00, .6564E+00, .6189E+00, .2501E+00, .2839E+00, .2622E+00, .2150E+00, .1774E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1995E+00, .1945E+00, .2148E+00, .2247E+00, .2277E+00, .2113E+00, .1789E+00, .1495E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .4943E+00, .1105E+00, .1137E+00, .2277E+00, .2247E+00, .1948E+00, .1789E+00, .1495E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1050F+00, .1069E+00, .2011E+00, .2054E+00, .2034E+00, .1827E+00, .1523E+00, .1099E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .5797E+00, .1049E+00, .1523E+00, .1827E+00, .2034E+00, .1523E+00, .1099E+00, .1050F+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1573E+00, .1716E+00, .1957E+00, .1860E+00, .1787E+00, .1605E+00, .1323E+00, .9452E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .3699E+00, .3431E+00, .1945E+00, .1323E+00, .1065E+00, .1787E+00, .1605E+00, .1250E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .1479E+00, .1669E+00, .1729E+00, .1792E+00, .1811E+00, .1429E+00, .1165E+00, .8257E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .4705E+00, .5267E+00, .5253E+00, .1811E+00, .1704E+00, .1681E+00, .1523E+00, .1250E+00</td>
<td></td>
</tr>
<tr>
<td>.1250E+04, .3779E+00, .3779E+00, .3779E+00, .3779E+00, .3779E+00, .3779E+00, .3779E+00, .1250E+00</td>
<td></td>
</tr>
</tbody>
</table>
Sample Case Plot: Simply Connected Region
### Sample Case Input: Single-Body Field

**TEST CASE: BODY-FITTED COORDINATE SYSTEM**

**SINGLE BODY: KARMAAN-TREFFZ AIRFOIL, 26 POINTS**

<table>
<thead>
<tr>
<th>K+T AIRFOIL #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 20 1 200 0 1 1 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0 0 0 1 1</td>
</tr>
<tr>
<td>2 1</td>
</tr>
<tr>
<td>1 1 27 1</td>
</tr>
<tr>
<td>3 1 27 0</td>
</tr>
<tr>
<td>2 1 20 4 1 20</td>
</tr>
</tbody>
</table>

| 1.8 0.0001 0.0001 5.0 0.0 0.0 0.0 26 |
| 0 0 0.0 0 0 0 0 0 0 |

| 499528 0.00031 |
| 498623 0.001402 |
| 496086 0.002760 |
| 486293 0.005552 |
| 470464 0.008509 |
| 390148 0.018591 |
| 269552 0.029966 |
| 122707 0.040790 |
| 0.34802 0.048453 |
| 187472 0.050524 |
| 321146 0.045592 |
| 423886 0.033739 |
| 486506 0.016526 |
| 502726 0.03820 |
| 471102 0.026435 |
| 396903 0.048642 |
| 287913 0.06101 |
| 153756 0.075299 |
| 495664 0.074531 |
| 144046 0.064352 |
| 282251 0.047498 |
| 395776 0.028262 |
| 471990 0.011299 |
| 487103 0.006809 |
| 496419 0.003040 |
| 499779 0.001426 |

| ETA 0 4 2 0.0 0.0 |
| 1 100.0 1.0 |
| 2 100.0 1.0 |
| 3 100.0 1.0 |
| 4 100.0 1.0 |
| 1 1 100.0 1.0 |
| 27 1 100.0 1.0 |

| XI 0 0 0 0.0 0.0 |
| 1 1 0 100.0 |
Sample Case Output: Single-Body Field

*************** Initial ***************

**INITIAL GUESS TYPE:**

**INITIAL GUESS DATA:**

**NUMBER OF POINTS:** 20
**NUMBER OF STEPS:** 1
**STEP IN ATTENTION OF INFINITY:**

**UNIFORM ACCELERATION PARAMETER USED:**
TEST CASE = NOUHFITTED CUCMULIATE SYSTEM
SINGLE NOUHF FITTED THEFFE AIPFOIL, 26 POINTS
NOUHF-FITTED CUCMULIATE SYSTEM
SHOTBEAM NOUHF, KRT AIRFOIL #1
FIELD PARAMETERS, NUMBER OF X-LINES = 27
NUMBER OF ETA-LINES = 20

ITERATION PARAMETERS SUM ACCELERATION PARAMETER = 1.000000
MAXIMUM NUMBER OF ITERATIONS ALLOWED = 200
ALLOWABLE ITERATION ERROR NUMBER #1 = 1.00000E+03
#2 = 1.00000E+03

NUMBER OF HIDDEN IN FIELD = 1
PLAT PARAMETERS
COPIES DESIRED = 1
LINEWEIGHT DESIRED = 0
PLAT SIZE IN PROPORTION = 0.00
WIDTH = 3.000

LUCICAL CONTROLS I LACC = VARIABLE ACCELERATION PARAMETER FIELD CONVERGENCE (INRELEVANT IF LCONTRUE)
LCPAC = ADDITION OF INHOMOGENEOUS TERM COMPLETED
LCUN = UNIFORM ACCELERATION PARAMETER

EXPONENTIAL DECAY WHO IS TACTIVATION =

--- ETA EQUATION WHO ---

ACTIVATION LINES

<table>
<thead>
<tr>
<th>J</th>
<th>J</th>
<th>EXP</th>
<th>DECAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.00000000</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>100.00000000</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>100.00000000</td>
<td>1.00000000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>100.00000000</td>
<td>1.00000000</td>
<td></td>
</tr>
</tbody>
</table>

ACTIVATION POINTS

<table>
<thead>
<tr>
<th>J</th>
<th>J</th>
<th>AMP</th>
<th>DECAY</th>
<th>LPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>100.00000000</td>
<td>1.00000000</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>100.00000000</td>
<td>1.00000000</td>
<td>0</td>
</tr>
</tbody>
</table>

1 STEPS IN ADDITION OF INHOMOGENEOUS TERM, INTERMEDIATE CONVERGENCE FACTOR = 100.00000000
1,00000000 OF INHOMOGENEOUS TERM
1,00000000 OF SUITM HUNDARY

----- MAXIMUM NUMBRS OF ITERATE CHANGES

<table>
<thead>
<tr>
<th>ITERM</th>
<th>AMPNUM</th>
<th>TANGENT</th>
<th>ACCLNUM</th>
<th>AVG,ACC, PAM,</th>
<th>LACC</th>
<th>LCPAC</th>
<th>LCUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00000E+01</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.00000E+00</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>1.07273E+01</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.00000E+00</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>1.17900E+01</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.00000E+00</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>1.5453E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.00000E+00</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>1.7622E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.00000E+00</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
TEST CASE - HYSWITCH COUPLATE SYSTEM
SINGLE BODY I KANHAN-THEFRIE ALGORITHM, 25 POINTS

FINAL VALUES

ITERATION CONVERGES,

INITIAL ITERATION ERROR NORMS: XI = 1.098E+01 YI = 3.663E+00 AT ITERATE #1
FINAL ITERATION ERROR NORMS: XI = 7.777E-04 YI = 9.349E-06 AT ITERATE #19
LOCATION OF MAXIMUM ITERATION ERROR XI = 11 YI = 25

*************** *******
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 
1 1.098E+01 1.097E+01 1.096E+01 1.095E+01 1.094E+01 1.093E+01 1.092E+01 1.091E+01 1.090E+01 1.089E+01 1.088E+01 1.087E+01 1.086E+01 1.085E+01 1.084E+01 1.083E+01 1.082E+01 1.081E+01 1.080E+01 1.079E+01 1.078E+01 1.077E+01 1.076E+01 1.075E+01
2 3.663E+00 3.662E+00 3.661E+00 3.660E+00 3.659E+00 3.658E+00 3.657E+00 3.656E+00 3.655E+00 3.654E+00 3.653E+00 3.652E+00 3.651E+00 3.650E+00 3.649E+00 3.648E+00 3.647E+00 3.646E+00 3.645E+00 3.644E+00 3.643E+00 3.642E+00 3.641E+00 3.640E+00
Sample Case Plot: Single-Body Field
Sample Case Input: Double-Body Field

<table>
<thead>
<tr>
<th>TEST CASE = BODY-FITTED COORDINATE SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOUBLE BODY FITTED, 26 POINTS ON EACH AIRFOIL-FLAP #1</td>
</tr>
<tr>
<td>69 20 2 200 0 1 1 0 0</td>
</tr>
<tr>
<td>1 3 1 0 0 10 0 1 1</td>
</tr>
<tr>
<td>4 2</td>
</tr>
<tr>
<td>1 1 14 =1</td>
</tr>
<tr>
<td>1 56 69 =1</td>
</tr>
<tr>
<td>1 22 48 =2</td>
</tr>
<tr>
<td>3 1 69 0</td>
</tr>
<tr>
<td>1 14 22 1 48 56</td>
</tr>
<tr>
<td>2 1 20 4 1 20</td>
</tr>
<tr>
<td>1,8 0.0001 0.0001 5.0 0.0 0.0 0.0 68</td>
</tr>
<tr>
<td>0 0 0.0</td>
</tr>
<tr>
<td>0 0</td>
</tr>
<tr>
<td>8.0 0.0 10.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>8.08147 = 1.62448</td>
</tr>
<tr>
<td>8.07780 = 1.62631</td>
</tr>
<tr>
<td>8.07061 = 1.62608</td>
</tr>
<tr>
<td>8.04591 = 1.61989</td>
</tr>
<tr>
<td>8.00795 = 1.60580</td>
</tr>
<tr>
<td>7.82146 = 1.52794</td>
</tr>
<tr>
<td>7.75461 = 1.40162</td>
</tr>
<tr>
<td>7.72146 = 1.24170</td>
</tr>
<tr>
<td>6.86409 = 1.06140</td>
</tr>
<tr>
<td>6.53046 = 0.87505</td>
</tr>
<tr>
<td>6.24771 = 0.69726</td>
</tr>
<tr>
<td>6.04009 = 0.54319</td>
</tr>
<tr>
<td>5.42603 = 0.42765</td>
</tr>
<tr>
<td>5.91634 = 0.36132</td>
</tr>
<tr>
<td>6.01508 = 0.35389</td>
</tr>
<tr>
<td>6.02048 = 0.39456</td>
</tr>
<tr>
<td>6.45928 = 0.49700</td>
</tr>
<tr>
<td>6.76123 = 0.64478</td>
</tr>
<tr>
<td>7.08095 = 0.83156</td>
</tr>
<tr>
<td>7.39231 = 1.04073</td>
</tr>
<tr>
<td>7.67046 = 1.24998</td>
</tr>
<tr>
<td>7.89221 = 1.43355</td>
</tr>
<tr>
<td>8.03601 = 1.56552</td>
</tr>
<tr>
<td>8.06312 = 1.59414</td>
</tr>
<tr>
<td>8.07858 = 1.61394</td>
</tr>
<tr>
<td>8.08167 = 1.62039</td>
</tr>
<tr>
<td>4.99528 = 0.00331</td>
</tr>
<tr>
<td>4.98623 = 0.01402</td>
</tr>
<tr>
<td>4.96086 = 0.02760</td>
</tr>
<tr>
<td>4.86293 = 0.05552</td>
</tr>
<tr>
<td>4.70464 = 0.08509</td>
</tr>
<tr>
<td>3.90148 = 0.18591</td>
</tr>
<tr>
<td>2.69552 = 0.29966</td>
</tr>
<tr>
<td>1.22707 = 0.40790</td>
</tr>
<tr>
<td>0.034802 = 0.048453</td>
</tr>
<tr>
<td>0.187472 = 0.050524</td>
</tr>
<tr>
<td>0.321140 = 0.045592</td>
</tr>
<tr>
<td>0.423886 = 0.033739</td>
</tr>
<tr>
<td>0.486560 = 0.016526</td>
</tr>
<tr>
<td>0.502726 = 0.038200</td>
</tr>
<tr>
<td>0.471102 = 0.024352</td>
</tr>
<tr>
<td>0.396905 = 0.046642</td>
</tr>
<tr>
<td>0.287915 = 0.061010</td>
</tr>
<tr>
<td>0.153756 = 0.075299</td>
</tr>
</tbody>
</table>
Sample Case Output: Double-Body Field

*************** Input **********************
IMAX, JMAX, NOUT, ITHM, IGES, IDISK, IIN, IINIT, IINF, IGEW 1 20 20 0 1 1 0 0 0
IPLT, IPLTH, ICOPY, LICH, LINIT, LNUMM, LNUM, LSKIP1, LSKIP2 1 3 1 0 10 0 1 1
LNSDG, LNSLG 1 4 2

T1 (T2), M(1), M(2), R(S), R(T), XFIN, AFIN, XINF, AFINF, XIIF, AIIF 1
F00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000

IEV, INIT, M(10) 1 0 0 0
INFAC, INFACO 1 0
SIZE, MATLC, DIST, RX, RX2, TX, TX2 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000

*************** Initial Guess Type: 0 ***************

**About Segments**

L LABSU LR1 LR2 LBDY
1 1 1 1 1
2 1 3 3 1
3 1 2 2 2
4 1 1 1 1

**Remnant Segments**

L LABSU LR1 LR2 LISU LI1 LIZ
1 1 14 22 1 48 58
2 2 1 20 4 1 20

**Outer Boundary**

RADIUS = 0.00000000, INITIAL ANGLE = 0.00000000

NUMBER OF POINTS = 68, 1 STEPS IN ATTAINMENT OF INFINITY, INITIAL STEP (LINKAN CASE) = 1

UNIFORM ACCELERATION PARAMETER USED,
FIRST CASE = NON-FITTED LUMINOSITE SYSTEM
DUMMY MINT = 2 PER KAMON THEORETICAL INPUTS, 20 POINTS ON EACH
NON-FITTED LUMINOSITE SYSTEM
TRANSFUERED MINT, ART IMODE=FLAP=1
FIELD PARAMETERS, NUMBER OF ELEMENTS = 60
NUMBER OF ETABLELINES = 2

ITERATION PARAMETERS
SUM ACCELERATION PARAMETER = 1.00000
MAXIMUM NUMBER OF ITERATIONS ALLOWED = 200
ALLOWABLE ITERATION ERROR NUMBER = 1.00E-08

NUMBER OF OUTPUTS IN FILLU = 4
PLUT PARAMETERS: CUPID HELIX=1
LINESLIT DEFILED = 0
PLUT SIZE IN CASECTION = 8.4000
MATU = 8.4000

LOGICAL CONTROLS
1 CALL = VARIABLE ACCELERATION PARAMETER FIELD CONVERGED (IMRELEVANT IF LECONTRUE)
LCFAC = ADDITION OF INHOMOGENEOUS TERM COMPLETED
LCON = UNIFORM ACCELERATION PARAMETER

EXPONENTIAL UDECAY =
= ATTACAN =

--- ETA EQUATION ---

ATTACAN LINES

<table>
<thead>
<tr>
<th>J</th>
<th>A</th>
<th>DENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
</tr>
<tr>
<td>2</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
</tr>
<tr>
<td>3</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
</tr>
<tr>
<td>4</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
</tr>
</tbody>
</table>

ATTACAN POINTS

<table>
<thead>
<tr>
<th>J</th>
<th>J</th>
<th>A</th>
<th>DENT</th>
<th>LFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>0</td>
</tr>
</tbody>
</table>

3 STEPS IN ADDITION OF INHOMOGENEOUS TERM, INTERMEDIATE CONVERGENCE FACTOR = 100.00000000

--- MAXIMUM NUMBER OF ITERATE CHANGES ---

<table>
<thead>
<tr>
<th>ITERATE</th>
<th>XNOM</th>
<th>ALLOUM</th>
<th>AVAXXM</th>
<th>AVHAXXMAXX</th>
<th>LACC</th>
<th>LCFAC</th>
<th>LCON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>6</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>7</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>8</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000200000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>74</td>
<td>45747r-0+</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>31806e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>25902e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>20507e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>11502e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>59403e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>51903e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>11547e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>31312e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>11102e-01</td>
<td>1.00000</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No text content extracted from the image.
121
Sample Case Plot: Double-Body Field
VII. INSTRUCTIONS FOR USE - SCALE FACTORS

After a coordinate system has been generated, the "scale factors" for use in the solution in any partial differential equation transformed to the rectangular transformed plane are generated and written on a disk file by the main program FATCAT with its subroutine ABG, AMXMN, PARA1, PARA2, PARA3, and WRDATA. Instructions for use are included in the listing of FATCAT.

Core must be set to zero at load time.

Dimensions. The standard program allows a maximum field size of 70 $\xi$ lines and 60 $\eta$ lines and requires a core size of 201,000 words for the Langley Research Center's CDC 6000 Series Computer System.

Input Parameters. Explanation found with the sample test data, and program listing.

Files. This program requires 2 essential files:

TAPE 1 - input tape - generated as TAPE10 by Program TOMCAT.

TAPE10 - disk on which the factors are to be written.
Scale Factors

Program FATCAT
PROGRAM FATCAT(INPUT,OUTPUT,TAPE10,TAPE11)
C
********** MISSISSIPPI STATE 2-D COORDINATE TRANSFORMATION **********
C
C     COORDINATE SYSTEM: SCALE FACTORS: ALPH, BETA, GAMMA, SIGMA, TAU,
C     JACOBIAN, DX/OXI, DX/DETA,
C     DY/OXI, AND DY/DETA,
C
C
C *********************************************************************
C
DIMENSION X(70,60), Y(70,60), C(10,70,60), CB(10,70,4)
DIMENSION C1(8),C2(8),LSID(6),LM1(6),LR2(6),LBDY(6),
     LRBD(16), LRI(6), L92(6), LISID(6), L11(6), L12(6), LTYPE(6), LSBS(6)
DATA NDIM,NDIM1,NDIMX,70,60,70/
C
C********************************************************************
C
C
C     THE COORDINATE SYSTEM IS READ FROM DISK(TAPE1) IN THE SAME
C     FORMAT USED BY TOMCAT TO WRITE ON DISK; THE SCALE FACTORS
C     ARE WRITTEN ON DISK(TAPE10) IN ONE OF THE FOLLOWING FORMATS,
C     SPECIFIED BY IFORM.
C
C *** FORMAT #1 (IFORM=1) ***
C
WRITE(10,1) C1
WRITE(10,1) C1
WRITE(10,1) IMAX,JMAX,NNSEG,NNSEG2,ISRSEG,ISBDY
WRITE(10,1) (LSID(L),LM1(L),LR2(L),LBDY(L),LSBS(L),
     1 L=1,NNSEG)
WRITE(10,1) (LSID(L),LM1(L),LR2(L),LISID(L),L11(L),L12(L),
     1 LTYPE(L),LM1,NNSEG)
DO 2 JMAX=2
  2 WRITE(10,1) (((N,J),N=1,10),IMAX=2)
DO 3 JMAX=4
  3 WRITE(10,1) (((CB(N,J),N=1,10),IMAX=MAX(IMAX,JMAX)
C
C     THE ARRAY C CONTAINS THE FACTORS FROM THE SECOND POINT TO THE
C     PENULTIMATE POINT, ON SECOND ROW TO THE PENULTIMATE ROW.
C     THE ARRAY CB CONTAINS THE FACTORS ON THE RECTANGULAR BOUNDARY,
C     J=1,2,3,4 IN NC CORRESPONDING RESPECTIVELY, TO THE BOTTOM,
C     LEFT, TOP, RIGHT SIDES, ON EACH SIDE THE POINTS RUN FROM 1 TO
C     EITHER IMAX OR JMAX, AS APPROPRIATE.
C
C *** FORMAT #2 (IFORM=2) ***
C
WRITE(10,1) C1
WRITE(10,1) C2
WRITE(10,1) IMAX,JMAX,NNSEG,NNSEG2,ISRSEG,ISBDY
WRITE(10,1) (LSID(L),LM1(L),LR2(L),LBDY(L),LSBS(L),
     1 L=1,NNSEG)
WRITE(10,1) (LSID(L),LM1(L),LR2(L),LISID(L),L11(L),L12(L),
     1 LTYPE(L),LM1,NNSEG)
DO 2 JMAX=4
  2 WRITE(10,1) (((C(N,J),N=1,10),IMAX=1,MAX(IMAX,JMAX)
C
C     THE ARRAY C CONTAINS THE FACTORS FROM THE FIRST POINT TO THE
C     LAST POINT, ON THE FIRST ROW TO THE LAST ROW.
C
C ***
C THE FACTORS CORRESPOND TO THE INDEX N AS FOLLOWS:
C
C N=1 JACOBIAN
C N=2 ALPHA
C N=3 BETA
C N=4 GAMMA
C N=5 SIGMA
C N=6 TAU
C N=7 DX/DXI
C N=8 DY/DXI
C N=9 DY/DETA
C N=10 dy/dxi
C
C ********************************************************** INPUT DATA *********************************************************
C
C *** CARD I WRIT1,WRIT2,IFORM = FORMAT(315)
C
C IWRIT1 = 0 DON'T PRINT COORDINATE SYSTEM FROM WHICH FACTORS ARE CALCULATED.
C >0 PRINT COORDINATE SYSTEM.
C
C IWRIT2 = 0 DON'T PRINT FACTORS.
C >0 PRINT FACTORS.
C
C IFORM = FILE STORAGE FORMAT Control, (SEE ABOVE)
C
C **********************************************************
C
C READ INPUT DATA
C
WRITE (6,180)
N1=1
REWIND N1
READ (5,110) IWRIT1, IWRIT2, IFORM
WRITE (6,190) IWRIT1, IWRIT2, IFORM
IF (IFORM.GT.1.ORIFORM.GT.2) WRITE (6,170)
IF (IFORM.GT.1) STOP 1

C READ COORDINATE SYSTEM
C
READ (N1,101) C1
READ (N1,101) C2
READ (N1,110) IMAX, JMAX, NRSEG, NRSEG, L1SEG, NRDY
ITEST=IMAX-2
JTEST=JMAX-2
IF (IFORM.EQ.2) ITEST=IMAX
IF (IFORM.EQ.2) JTEST=JMAX
IF (ITEST.GT.NDIM) WRITE (6,130)
IF (JTEST.GT.NDIM) WRITE (6,140)
IF (ITEST.GT.NDIM.OR.JTEST.GT.NDIM) STOP 2
IF (MAXO(IMAX,JMAX),GT,NDIMX) WRITE (6,160)
IF (MAXO(IMAX,JMAX),GT,NDIMX) STOP 3
READ (N1,102) (LB2ID(L),LR1(L),LR2(L),LBDY(L),LBEN(L),L=1,NRSEG)
READ (N1,103) (LB2J(L),LR1(L),LR2(L),LBTID(L),LII(L),LII(L),
1LYPE(L),L=1,NRSEG)
READ (N1,104) (X(I,J),IM1,IMAX,J=1,IMAX),((Y(I,J),IM1,IMAX),
1JM1,JMAX)
C PRINT LABEL
  WRITE (6,100) C1
  WRITE (6,100) C2
  WRITE (6,120) IMAX,JMAX
C CALCULATE FACTORS
  JMAX1=JMAX+1
  IMAX1=IMAX+1
  IMXMAX(1,IMAX,JMAX)
  IMX2=THMAX+2
  JM1=2*JMAX+2
  CALL ABG (C,X,Y,NM,XM,X,M,JAX,JUAX,?X,LI,TE,LB, LIST, LID, L_2, L_{2 Y }, LTYPE, LSEF, NDIMX, NSED)
C PRINT X AND Y FIELDS
  IF (IRT1.EQ.0) GO TO 10
  CALL *DATA (X,IMAX,JMAX,1,2,NDIM)
  CALL *DATA (Y,IMAX,JMAX,3,4,NDIM)
C WRITE FACTORS TO DISK
  10 GO TO (20,50), IFOR
C *** FORMAT #1 ***
  20 WRITE(10,101) C1
    WRITE(10,101) C2
    WRITE(10,101) IMAX,JMAX,NRBEG,NRSEG,LSEG,NBYD
    WRITE(10,102) (LSENT(L),LR1(L),LR2(L),LRHNY(L),LBEN(L),LWI1,WRSEG)
    WRITE(10,103) (LSENT(L),LR1(L),LR2(L),LIS1(L),LII(L),LIZ2(L),
      LTYPE(L),L1I1,WRSEG)
    DO 30 J=1,JMXM2
      30 WRITE(10,104) ((C(N,I,J),NBBEG,NRBEG),I=1,IMAX)
    DO 50 J=1,NB
    40 WRITE(10,104) (CB(N,I,J),NBBEG,NRBEG),I=1,NDIMX
  STOP 0101
C *** FORMAT #2 ***
  50 WRITE(10,101) C1
    WRITE(10,101) C2
    WRITE(10,101) IMAX,JMAX,NRBEG,NRSEG,LSEG,NBYD
    WRITE(10,102) (LSENT(L),LR1(L),LR2(L),LRHNY(L),LBEN(L),LWI1,WRSEG)
    WRITE(10,103) (LSENT(L),LR1(L),LR2(L),LIS1(L),LII(L),LIZ2(L),
      LTYPE(L),L1I1,WRSEG)
    DO 60 J=1,JMXM2
      DO 60 II=1,IMAX
    60 DO 70 J=1,10
      JJ=JMXM1+JJ
      III=IMAXII
      JJ=JMAXJ
      III=IMAXII
      C(N,I,J)+C(N,III,JJJ)
    DO 70 I=1,IMAX
    DO 70 NI=1,10
        C(N,I,J)+C(N,II)
DIMENSION X(NDIM,1), Y(NDIM,1), CF(10,NDIM,NDIM1), C(10,NDIMX,4)
DIMENSION LBID(1), LBI(1), LB2(1), LB3(1), LB4(1), LBI(1), LR1(1), LRI(1),
1(1), L2ID(1), L2ID(1), L2ID(1), L2ID(1), L2ID(1), L2ID(1), L2ID(1), L2ID(1),
DIMENSION FAC(10), SIDE(4)
INTEGER FAC, SIDE
REAL JCB
DATA FAC /6HJACOBN,6HALPHA,6HETA,6HBETA,6HAGAMMA,6HBIGMA,6HTAU
6HMX,6HY,6XI,6HY,6XI,6HY,6XI,
DATA SIDE/6HBOTTOM,6HLEFT,6HTOP,6HRIGHT/
GO TO 260
C
C**** BODY SEGMENTS ****
C
10 DO 100 LB1,NBSEG
11=LB1(L)
2=LB2(1)
3=LB1(L)
13=II+1
14=II+1
15=II+1
16=II+1
STOP 0102
C
100 FORMAT(1X,RA10)
101 FORMAT(8A10)
102 FORMAT(5I5)
103 FORMAT(T75)
104 FORMAT(16,8)
105 FORMAT (16I5)
120 FORMAT (*OFIELD; IMAX =,I4,5X,JMAX =,I4)
130 FORMAT (*OHi= E n oW =*7轨道IMAX TOO LARGE, INCREASE NDim
140 FORMAT (*OHi= ERROR =*7轨道IMAX TOO LARGE, INCREASE NDim
150 FORMAT (*OHi= ERROR =*7轨道IMAX TOO LARGE, INCREASE NDim
160 FORMAT (*OHi= ERROR =*7轨道IMAX TOO LARGE, INCREASE NDim
2NATION/**
3* and NDIMX in DATA STATEMENT,**
170 FORMAT (*OHi= ERROR =*7轨道IMAX TOO LARGE, INCREASE NDim
180 FORMAT (1HI/1)
C
END

SUBROUTINE ABG (CF,X,Y,NDIM,IMAX,JMAX,IMX41,JMX41,IMX41,PNRSEG,
1LIBEG,CLBB1D,LBI,LB2,LB3,LBDY,LRSTIN,LR1,L2,LISID,L2ID,L2ID,L5TYPE,
2LB1N,NDIMX,NBSEG,NDIM1)
C
******* SCALE FACTORS *******
C
C
C
C
C
C
C(5,1C,JC)=BIG
C(6,1C,JC)=TAU
C(7,1C,JC)=XXI
C(8,1C,JC)=XETA
C(9,1C,JC)=YXI
C(10,1C,JC)=YETA
CALL PARA3 (X,Y,1,12,NDIM,XXI,YXI,1,12,1,2,1,2,3,1,2,3,1,2,3,1,2,XETA,YETA,1,12,
1,12=1,1,12=2,1,=1,RETA,GAMA,IGOTO,JCB,ALPHA)
J=12
IC=12
C(1,1C,JC)=JCB
C(2,1C,JC)=ALPHA
C(3,1C,JC)=RETA
C(4,1C,JC)=GAMA
C(5,1C,JC)=SIG
C(6,1C,JC)=TAU
C(7,1C,JC)=XXI
C(8,1C,JC)=XETA
C(9,1C,JC)=YXI
C(10,1C,JC)=YETA
DO 50 I=1,14
CALL PARA3 (X,Y,1,12,NDIM,XXI,YXI,1,T,2,1,3,1,XETA,YETA,1,1,1,0,
1,0,1,1=1,1,0,RETA,GAMA,IGOTO,JCB,ALPHA)
IC=1
C(1,1C,JC)=JCB
C(2,1C,JC)=ALPHA
C(3,1C,JC)=BETA
C(4,1C,JC)=GAMA
C(5,1C,JC)=SIG
C(6,1C,JC)=TAU
C(7,1C,JC)=XXI
C(8,1C,JC)=XETA
C(9,1C,JC)=YXI
C(10,1C,JC)=YETA
CONTINUE
50 GO TO 100
C**** TOP
60 CALL PARA3 (X,Y,1,12,NDIM,XXI,YXI,1,1,1,JMAX,1,1+1,JMAX,1,1+2,JMAX
1,XETA,YETA,1,12,JMAX,1,12=1,12=2,1,=1,RETA,GAMA,IGOTO,JCB,ALPHA)
J=JMAX
I=11
JC=3
IC=11
C(1,1C,JC)=JCB
C(2,1C,JC)=ALPHA
C(3,1C,JC)=BETA
C(4,1C,JC)=GAMA
C(5,1C,JC)=SIG
C(6,1C,JC)=TAU
C(7,1C,JC)=XXI
C(8,1C,JC)=XETA
C(9,1C,JC)=YXI
C(10,1C,JC)=YETA
CALL PARA3 (X,Y,1,12,NDIM,XXI,YXI,1,12,JMAX,1,12=1,12=2,1,=1,RETA,GAMA,IGOTO,JCB,ALPHA)
J=JMAX
I=12
IC=12
C(1,1C,JC)=JCB
CALL PARA3 (X, Y, I, JMAX, NDIM, XXI, YXI, I=I-1, JMAX, I+1, 1, JMAX, I)

CONTINUE
132

1ETA, YETA, IMAX, I=1, 0, 0, IMAX, I=1, 0, HFTA, GAMA, IGOTN, JCR, ALPHA)  421
1C=I
1C(1, IC, JC) = JCR  422
1C(2, IC, JC) = ALPH A  423
1C(3, IC, JC) = HFTA  424
1C(4, IC, JC) = GAMA  425
1C(5, IC, JC) = SIG  426
1C(6, IC, JC) = TAU  427
1C(7, IC, JC) = XXI  428
1C(8, IC, JC) = YETA  429
1C(9, IC, JC) = YXI  430
1C(10, IC, JC) = YETA  431
90 CONTINUE  432
100 CONTINUE  433

1C**** REENTRANT SEGMENTS ****  434
1C
1IF (NRSEG, EQ, 0) GO TO 245  435
1DO 240 L=1, NRSEG  436
1IS=LR1(L)+1  437
1IS=LR2(L)+1  438
1IS=L11(L)+1  439
1IS=L12(L)+1  440
1IGOTO=TYPE(L)  441
1GO TO (110, 130, 150, 170, 190, 210), IGOTO  442

1C**** ONE ON BOTTOM, ONE ON TOP  443
110 DO 120 I=15, 16  444
1J=1  445
1XXI=X(I+1,1)-X(I,1)*0.5  446
1YXI=Y(I+1,1)-Y(I,1)*0.5  447
1XETAM=X(I,2)-X(I,1)*0.5  448
1YETAM=Y(I,2)-Y(I,1)*0.5  449
1I2=I+1  450
1J2=J+1  451
1CALL PARA1 (XXI, YXI, J2, J1, XETAM, YETAM, I2, I1, XXIETAM, YXIETAM, XETAM, YETAM)  452
1CALL PARA2 (XXI, YXI, XETAM, YETAM, I2, I1, J2, J1, NIM)  453
1120 CONTINUE  454

1C**** BOTH ON BOTTOM  455
1130 DO 140 I=15, 16  456
1J=1  457
1II=I=I+15  458
1XXI=X(I+1,1)-X(I,1)*0.5  459
1YXI=Y(I+1,1)-Y(I,1)*0.5  460
1140 CONTINUE  461

1C**** BOTH ON BOTTOM  462
1150 DO 160 I=15, 16  463
1I=I  464
1II=I  465
1XXI=X(I+1,1)-X(I,1)*0.5  466
1YXI=Y(I+1,1)-Y(I,1)*0.5  467
1160 CONTINUE  468
YETA*(Y(I,J+1)*Y(I,J+1)) .eq. 0.5 541
I1=INX=1 542
I2=2 543
J1=J=1 544
J2=J+1 545
CALL PARA1 (X(I,J,J), I1, I2, J1, J, YETAS, I, J2, I, J1, XXIETA, Y 546
1XETA, I2, J2, J1, I1, I2, X, Y, J1, J, INX=1) 547
CALL PARA2 (X(I, J), YETAS, X(I, J), XXIETA, YXI 548
1ETA, ALPHA, GAMMA, BETA, SIG, TAU, JCR, I, J, INX=1) 549
JCR=2 550
IC=1 551
C(1, IC, JC)=JC 552
C(2, IC, JC)=ALPHA 553
C(3, IC, JC)=BETA 554
C(4, IC, JC)=GAMMA 555
C(5, IC, JC)=SIG 556
C(6, IC, JC)=TAU 557
C(7, IC, JC)=XXI 558
C(8, IC, JC)=YXI 559
C(9, IC, JC)=YETA 560
C(10, IC, JC)=YETA 561
CONTINUE 562
GO TO 210 563
C**** BOTH ON LEFT 564
190 DO 200 J=IS, 16 495
I=1 566
I=1=I=(J=IS) 567
XXI=(X(I,J)=X(I,J)) .eq. 0.5 568
YXI=(Y(I,J)=Y(I,J)) .eq. 0.5 569
XETA=(X(I,J+1)=X(I,J+1)) .eq. 0.5 570
YETA=(Y(I,J+1)=Y(I,J+1)) .eq. 0.5 571
I=II 572
I=1=1 573
J1=J+1 574
J2=J+1 575
CALL PARA1 (X(I,J,J), I1, I2, J1, I2, I1, I1, J1, X(I,J,J), YETAS, I, J2, I, J1, XXIETA, 576
1XETA, I2, J2, J1, I1, I2, I1, I1, X, Y, J1, J1, INX=1) 577
CALL PARA2 (X(I, J), YETAS, X(I, J), XXIETA, YXI 578
1ETA, ALPHA, GAMMA, BETA, SIG, TAU, JCR, I, J, INX=1) 579
JCR=2 580
IC=1 581
C(1, IC, JC)=JC 582
C(2, IC, JC)=ALPHA 583
C(3, IC, JC)=BETA 584
C(4, IC, JC)=GAMMA 585
C(5, IC, JC)=SIG 586
C(6, IC, JC)=TAU 587
C(7, IC, JC)=XXI 588
C(8, IC, JC)=YXI 589
C(9, IC, JC)=YETA 590
C(10, IC, JC)=YETA 591
CONTINUE 592
GO TO 210 593
C**** BOTH ON RIGHT 594
210 DO 220 J=IS, 16 595
I=IMAX 596
I=I=I=(J=IS) 597
XXI=(X(I, J), I1, I1, IMX=1, J) .eq. 0.5 598
YXI=(Y(I, J), I1, I1, IMX=1, J) .eq. 0.5 599
XETA=(X(I, IMAX, J+1)=X(I, IMAX, J+1)) .eq. 0.5 600
VFT = f(y(I_{MAX}, J+1) - y(I_{MAX}, J=1)) \times n, 5

1 = I_{IT}
2 = T_{MX} = 1
J_{IM} = 1
J_{M} = J + 1
CALL PAHA1 (XXI, YXI, I_{2}, I_{1}, J_{2}, J_{1}, YETA, YETAS, I_1, J_1, XXIETA, YETAF, YETAF0, I_{11}, I_{12}, J_{11}, J_{12}, YETA, YETAS, XXIETA, YETAF, YETAF0, I_{11}, J_{12})

CALL PAHA2 (XXI, YXI, ETA, ETA, XXI, YXI, ETA, ETA, XXIETA, YETAF, YETAF0, ETA, ETA, XXIETA, YETAF, YETAF0, ETA, ETA, XXIETA, YETAF, YETAF0)

J_{C} = 1
C/J
C(1, J, C) = JCB
C(2, J, C) = XCB
C(3, J, C) = ETA
C(4, J, C) = GAMA
C(5, J, C) = SIG
C(6, J, C) = TAU
C(7, J, C) = XXI
C(8, J, C) = ETA
C(9, J, C) = YXT
C(10, J, C) = YETA

220 CONTINUE
230 CONTINUE
240 CONTINUE
245 DO 246 N = 1, 10

TEM = C(N, 1, 1) + C(N, 1, 2)
C(N, 1, 1) = TEM
C(N, 1, 2) = TEM
TEM = C(N, J_{MAX}, 2) + C(N, 1, 3)
C(N, J_{MAX}, 2) = TEM
C(N, 1, 3) = TEM
TEM = C(N, J_{MAX}, 3) + C(N, J_{MAX}, 4)
C(N, J_{MAX}, 3) = TEM
C(N, J_{MAX}, 4) = TEM
TEM = C(N, 1, 4) + C(N, I_{MAX}, 1)
C(N, 1, 4) = TEM
C(N, I_{MAX}, 1) = TEM

244 CONTINUE
IF (TMRT2, EQU, 0) GO TO 290
WRITE (6, 330)
DO 250 J = 1, 4
WRITE (6, 340) SIDE(J)
IF (J, EQ, 1) IMAXX1 = IMAX
IF (J, EQ, 3) IMAXX1 = IMAX
IF (J, EQ, 2) IMAXX1 = JMAX
IF (J, EQ, 4) IMAXX1 = JMAX
DO 250 N = 1, 10
WRITE (6, 320) FAC(N), (C(N, I, J), I = 1, I_{MAXX1})
250 CONTINUE
GO TO 290
C
C *** FIELD ***
C
260 CONTINUE
DO 270 J = 2, J_{MAXM1}
JC = J
JH = J
JP1 = J + 1
DO 270 IM = 2, I_{MAXM1}
IC = IM
IP1 = IM + 1

CONTINUE

FUNCTION AMXN (NOPT, A, AX, I, J, IX, JX)

C

300 FORMAT (*0---- SCALE FACTORS -----*)
310 FORMAT (// * J /*, I3/*)
320 FORMAT (*0, A6//("8E15.8")
330 FORMAT (// * BOUNDARY*)
340 FORMAT (//2H *", A6")

END
**SUBROUTINE PARAl (XI,YI,I1,J1,I2,J2,X2,Y2,I3,J3,I4,J4,X12,Y12,15,J15,16,J6,17,J7,18,J8,X,Y,I,J,NDI)**

**C**

```
******
DIMENSION X(NDIM,1), Y(NDIM,1)
******
```

```
X1=2.0*X(I,J)+X(I2,J2)
Y1=2.0*Y(I,J)+Y(I2,J2)
X2=2.0*X(I,J)+X(I4,J4)
Y2=2.0*Y(I,J)+Y(I4,J4)
X12=2.0*(X(I5,J5)=X(I8,J6)*X(I7,J7)=X(I8,J8))
Y12=2.0*(Y(I5,J5)=Y(I8,J6)*Y(I7,J7)=Y(I8,J8))
RETURN
C
END
```

**SUBROUTINE PARa2 (XXI,YXI,XETA,YETA,XXIS,YXIS,XETAS,YETAS,XXIETA,YXIETA,ALPHA,GAMA,BETA,TAU,JCB,I,J,NDI)**

**C**

```
******
REAL JCB
******
```

```
ALPHA**2*YETA**2
BETA**2*XXI**2*XETA+YXI**2*YETA
GAMMA**2*XXIS**2*YXIS*YXI
JCB*XXI*YETA=XETA*YXI
DX=ALPHA*XXIS**2+BETA**2*YXI**2
```

**C**
DY=ALPHA*YXI*2+.0*ETA*YIETA*GAMA*YETAR
SIG=D(Y*YXI)/JCB
TAU=(D*Y*ETA)/JCB
RETURN
C
END

SUBROUTINE PARAS (X,Y,I,J,NDIM,XXI,YXI,YI,YII,J1,J2,J3,YETA,YTEA)
1,J4,J5,J6,Jh,K1,K2,ETA,GAMA,IGNOM,JCR,ALPHA)
C
C *************** ALPHA, BETA, GAMMA, JACOBIAN ***************
C *
C *************** DIMENSION X(NDIM,1), Y(NDIM,1) ***************
C
C*****
IF (K2,LEQ,0) GO TO 10
XXI0.5*(X(I5,J3)=4.0*X(J2,J2)+3.0*X(J1,J1))=K1
YXI0.5*(Y(I5,J3)=4.0*Y(J2,J2)+3.0*Y(J1,J1))=K1
GO TO 20
10 XXIM0.5*(X(16,J3)=X(I1,J1))
YXM0.5*(Y(I6,J3)=Y(I1,J1))
20 IF (K2,LEQ,0) GO TO 30
YETA0.5*(Y(16,Jh)=4.0*Y(J5,J5)+3.0*Y(J4,J4))=K2
YETAM0.5*(Y(16,Jh)=4.0*Y(J5,J5)+3.0*Y(J4,J4))=K2
GO TO 40
30 YETAM0.5*(Y(16,Jh)=Y(I4,J4))
YETAM0.5*(Y(I6,Jh)=Y(I4,J4))
40 CONTINUE
ALPHA=YETA**2*YTEA**2
BETA*XXI*YETA+YXI*YETA
GAMA*XXI**2*YXI**2
JCR*XXI*YETA=YETA*YXI
RETURN
C
END

SUBROUTINE HRDATA (XI,IMAX,JMAX,I1,I2,NDIM)
C
C *************** PRINT SOLUTION ***************
C *
C *************** DIMENSION CODE(4) ***************
C
C DIMENSION X(NDIM,1)
C DATA CODE(4)
C
C****
WRITE (6,20) (CODE(I),I=I1,I2)
DO 10 J=1,JMAX
WRITE (6,30) J
10 WRITE (6,40) (X(I,J),I=I1,IMAX)
RETURN
C
C
20 FORMAT (1H1,20X,2A6//)
30 FORMAT (5x,1I8*14)
40 FORMAT (4x,10E11.5)
C
FNL
SAMPLE CASE INPUT: SINGLE-BODY FIELD

Input for Program FATCAT requires only one card and is described beginning at line number 76 in FATCAT listing. For the sample case computed:

```
1 1 2
```

Disk file 1 needed as input is TAPE 10 saved from the TOMCAT program.
Sample Case Output: Single-Body Field

INPUT 1  INPUT 2  INPUT 3  INPUT 4
TEST CASE = BODY+FITTED COORDINATE SYSTEM
SINGLE BODY = AIRFOIL, 12 POINTS
FIELD 1 MAX = 27  MAX = 20

**** SCALE FACTORS ****
J = 2

JACOBA

1.0256213e+03
4.6150387e-02
6.043732e-02
1.0307133e+03

1.3555100e+03
1.952579e+02
1.937534e+02
1.934794e+03

1.2675011e+03
1.007172e+03
2.014874e-01
2.125942e-01

1.3249090e+03
6.197753e+03
2.196154e+01
3.103198e+00

1.2586420e+03
1.507381e+03
2.775070e+03
1.006390e+03

1.7187530e-08
1.8052697e-07
1.6238420e-02
1.0794205e-02

1.8794205e-02
1.1033900e-01
1.4285714e+00
6.2895180e+00

1.6817240e-02
1.0701170e+01
1.2126250e-01
9.422300e+00

1.5759495e+01
1.1362501e-01
1.719753e-03
1.4887200e-01

1.3495100e+03
7.976473e-07
2.8741153e+05
5.700419e+05

1.3555100e+03
1.952579e+02
1.937534e+02
1.934794e+03

1.2675011e+03
1.007172e+03
2.014874e-01
2.125942e-01

1.3249090e+03
6.197753e+03
2.196154e+01
3.103198e+00

1.2586420e+03
1.507381e+03
2.775070e+03
1.006390e+03

1.7187530e-08
1.8052697e-07
1.6238420e-02
1.0794205e-02

1.8794205e-02
1.1033900e-01
1.4285714e+00
6.2895180e+00

1.6817240e-02
1.0701170e+01
1.2126250e-01
9.422300e+00

1.5759495e+01
1.1362501e-01
1.719753e-03
1.4887200e-01

1.3495100e+03
7.976473e-07
2.8741153e+05
5.700419e+05
```
<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>+15470294X +384</td>
</tr>
<tr>
<td>ETA</td>
<td>0.0</td>
</tr>
<tr>
<td>GAMMA</td>
<td>0.0</td>
</tr>
<tr>
<td>SIGMA</td>
<td>0.0</td>
</tr>
<tr>
<td>TAU</td>
<td>0.0</td>
</tr>
<tr>
<td>X, X1</td>
<td>0.0</td>
</tr>
<tr>
<td>X, ETA</td>
<td>0.0</td>
</tr>
<tr>
<td>Y, ETA</td>
<td>0.0</td>
</tr>
<tr>
<td>Y, X1</td>
<td>0.0</td>
</tr>
<tr>
<td>X, X1</td>
<td>0.0</td>
</tr>
<tr>
<td>X, ETA</td>
<td>0.0</td>
</tr>
<tr>
<td>Y, ETA</td>
<td>0.0</td>
</tr>
<tr>
<td>Y, X1</td>
<td>0.0</td>
</tr>
</tbody>
</table>
```


<table>
<thead>
<tr>
<th>Figure</th>
<th>IMAAX</th>
<th>JMAAX</th>
<th>NBSEG</th>
<th>NRSEG</th>
<th>LBSID</th>
<th>LB1</th>
<th>LB2</th>
<th>LEDY</th>
<th>LRSID</th>
<th>LR1</th>
<th>LR2</th>
<th>LISID</th>
<th>L11</th>
<th>L12</th>
<th>NINF</th>
<th>Body #1</th>
<th>Body #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4a, 14</td>
<td>37</td>
<td>20</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>37</td>
<td>1</td>
<td></td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>4</td>
<td>1</td>
<td>20</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>5, 15</td>
<td>81</td>
<td>21</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>66</td>
<td>26</td>
<td>56</td>
<td>1</td>
<td>1</td>
<td>26</td>
<td>4</td>
<td>1</td>
<td>66</td>
<td>1</td>
<td>80</td>
</tr>
<tr>
<td>16</td>
<td>71</td>
<td>16</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>51</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>3</td>
<td>1</td>
<td>31</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>17</td>
<td>56</td>
<td>31</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>31</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>3</td>
<td>3</td>
<td>21</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>8, 18</td>
<td>51</td>
<td>36</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>11</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>16</td>
<td>50</td>
<td>30/270-U</td>
</tr>
<tr>
<td>9, 19</td>
<td>31</td>
<td>46</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>11</td>
<td>4</td>
<td>1</td>
<td>11</td>
<td>30</td>
<td>30/270-U</td>
</tr>
<tr>
<td>10, 20</td>
<td>71</td>
<td>26</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>71</td>
<td>2</td>
<td>1</td>
<td>21</td>
<td>4</td>
<td>1</td>
<td>71</td>
<td>30</td>
<td>70/270-U</td>
</tr>
</tbody>
</table>
### TABLE 1 Continued

<table>
<thead>
<tr>
<th>Figure</th>
<th>IMA</th>
<th>JMA</th>
<th>NRSEG</th>
<th>NRSEG</th>
<th>LBSID</th>
<th>LB1</th>
<th>LB2</th>
<th>LBDY</th>
<th>LRSID</th>
<th>LR1</th>
<th>LR2</th>
<th>LISID</th>
<th>LI1</th>
<th>LI2</th>
<th>NINF</th>
<th>Body #1</th>
<th>Body #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11, 21</td>
<td>101</td>
<td>11</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>101</td>
<td>86</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>11</td>
<td>66</td>
<td>3</td>
<td>4</td>
<td>11/86</td>
<td>30/100/270-U</td>
</tr>
<tr>
<td>12, 22</td>
<td>71</td>
<td>36</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>51</td>
<td>21</td>
<td>71</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>11</td>
<td>21</td>
<td>4</td>
<td>71/270-L</td>
<td>30/30/90-U</td>
</tr>
<tr>
<td>13, 23</td>
<td>101</td>
<td>21</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>101</td>
<td>101</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>11</td>
<td>21</td>
<td>4</td>
<td>21/270-L</td>
<td>30/30/90-U</td>
</tr>
</tbody>
</table>

**NOTE:** The first number in the body name is the number of points on the body. The second number is the angle (counterclockwise from the x-axis) of the first point. The letter indicates a location above (U) or below (L) the x-axis.
TABLE 2

Effect of Initial Guess on Convergence

<table>
<thead>
<tr>
<th>Configuration</th>
<th>IGES:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0  1  2  3  -1 -4 -10 -20 -40 -100</td>
</tr>
<tr>
<td>(Fig. 4a)</td>
<td>68  NC  80  NC  NC  NC  NC  --  69  71</td>
</tr>
<tr>
<td>(Fig. 5)</td>
<td>75  NC  64  NC  100+ 100+  70  63  67  68</td>
</tr>
<tr>
<td>(Fig. 6)</td>
<td>NC  NC 1 NC  NC  NC  NC  NC  NC  NC  NC</td>
</tr>
<tr>
<td>(Fig. 7)</td>
<td>NC  NC  NC  NC1  NC  NC  NC  NC  NC  NC</td>
</tr>
<tr>
<td>(Fig. 8)</td>
<td>100+ 100+ 100+ NC  NC  100+ 100+  87  84  86</td>
</tr>
<tr>
<td>(Fig. 9)</td>
<td>100+ 100+ 100+ NC  NC  100+  NC  NC  NC  NC</td>
</tr>
<tr>
<td>(Fig. 10)</td>
<td>99  89  99  NC  NC  100+  91  84  82  86</td>
</tr>
<tr>
<td>(Fig. 11)</td>
<td>NC  100+ 100+ NC  NC  100+ 100  93  91  94</td>
</tr>
<tr>
<td>(Fig. 12)</td>
<td>NC  NC  NC  NC  NC  100+ 100+ 100+ 100+ 100+</td>
</tr>
<tr>
<td>(Fig. 13)</td>
<td>NC  NC  NC  100+ NC  NC  100+ 100+ 100+ 99</td>
</tr>
</tbody>
</table>

Legend: Numbers given are the number of iterations required for convergence to 0.0001.

NC indicates no tendency toward convergence in 100 iterations.

100+ indicates a converging solution at 100 iterations.

Notes: 1IGES = 1 gave convergence for the field shown in Figure 6.

2IGES = 4 (Guess 3 without division by total number of boundary points) gave convergence for the field shown in Figure 7.
**TABLE 3**

Input Parameters for Basic Single-Body Field Used in Optimum Acceleration Parameter Studies

Case 1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAX</td>
<td>67</td>
</tr>
<tr>
<td>JMAX</td>
<td>40</td>
</tr>
<tr>
<td>NBDY</td>
<td>1</td>
</tr>
<tr>
<td>IGES</td>
<td>0</td>
</tr>
<tr>
<td>NBSEG</td>
<td>2</td>
</tr>
<tr>
<td>NRSEG</td>
<td>1</td>
</tr>
<tr>
<td>LBSID</td>
<td>1</td>
</tr>
<tr>
<td>LB1</td>
<td>1</td>
</tr>
<tr>
<td>LB2</td>
<td>67</td>
</tr>
<tr>
<td>LBDY</td>
<td>1</td>
</tr>
<tr>
<td>LRSID</td>
<td>2</td>
</tr>
<tr>
<td>LR1</td>
<td>1</td>
</tr>
<tr>
<td>LR2</td>
<td>40</td>
</tr>
<tr>
<td>LISID</td>
<td>4</td>
</tr>
<tr>
<td>LI1</td>
<td>1</td>
</tr>
<tr>
<td>LI2</td>
<td>40</td>
</tr>
<tr>
<td>R(1)</td>
<td>varied</td>
</tr>
<tr>
<td>R(2)</td>
<td>0.0001</td>
</tr>
<tr>
<td>R(3)</td>
<td>0.0001</td>
</tr>
<tr>
<td>YINFIN</td>
<td>10.0</td>
</tr>
<tr>
<td>AINFIN</td>
<td>0.0</td>
</tr>
<tr>
<td>XOINF</td>
<td>0.0</td>
</tr>
<tr>
<td>YOINF</td>
<td>0.0</td>
</tr>
<tr>
<td>NINF</td>
<td>66</td>
</tr>
</tbody>
</table>
### TABLE 3 Continued

<table>
<thead>
<tr>
<th>IEV</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R(10)</td>
<td>0.0</td>
</tr>
<tr>
<td>INFAC</td>
<td>0</td>
</tr>
<tr>
<td>INFACO</td>
<td>0</td>
</tr>
<tr>
<td>ATYP</td>
<td>ETA</td>
</tr>
<tr>
<td>ITYP</td>
<td>0</td>
</tr>
<tr>
<td>NLN</td>
<td>1</td>
</tr>
<tr>
<td>NPT</td>
<td>0</td>
</tr>
<tr>
<td>DEC</td>
<td>0.0</td>
</tr>
<tr>
<td>AMPFAC</td>
<td>0.0</td>
</tr>
<tr>
<td>JLN</td>
<td>1</td>
</tr>
<tr>
<td>ALN</td>
<td>1000.0</td>
</tr>
<tr>
<td>DLN</td>
<td>1.0</td>
</tr>
<tr>
<td>ATYP</td>
<td>0</td>
</tr>
<tr>
<td>ITYP</td>
<td>0</td>
</tr>
<tr>
<td>NLN</td>
<td>0</td>
</tr>
<tr>
<td>NPT</td>
<td>0</td>
</tr>
<tr>
<td>DEC</td>
<td>0.0</td>
</tr>
<tr>
<td>AMPFAC</td>
<td>0.0</td>
</tr>
<tr>
<td>IFAC</td>
<td>0</td>
</tr>
<tr>
<td>EFAC</td>
<td>100.0</td>
</tr>
<tr>
<td>Case</td>
<td>IMAX</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>97</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>60</td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>37</td>
</tr>
<tr>
<td>11</td>
<td>37</td>
</tr>
<tr>
<td>12</td>
<td>97</td>
</tr>
<tr>
<td>13</td>
<td>97</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>60</td>
</tr>
<tr>
<td>17</td>
<td>60</td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 4 Continued

<table>
<thead>
<tr>
<th>Case</th>
<th>IMAX</th>
<th>JMAX</th>
<th>LB2</th>
<th>LR2</th>
<th>LI2</th>
<th>YINFIN</th>
<th>NINF</th>
<th>ALN</th>
<th>IFAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>37</td>
<td>20</td>
<td>37</td>
<td>20</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>23</td>
<td>97</td>
<td>20</td>
<td>97</td>
<td>20</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td>96</td>
</tr>
<tr>
<td>24</td>
<td>37</td>
<td>60</td>
<td>37</td>
<td>60</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>25</td>
<td>97</td>
<td>60</td>
<td>97</td>
<td>60</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td>96</td>
</tr>
<tr>
<td>26</td>
<td>37</td>
<td></td>
<td>37</td>
<td></td>
<td></td>
<td>5.0</td>
<td></td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>27</td>
<td>97</td>
<td></td>
<td>97</td>
<td></td>
<td></td>
<td>5.0</td>
<td></td>
<td></td>
<td>96</td>
</tr>
<tr>
<td>28</td>
<td>37</td>
<td></td>
<td>37</td>
<td></td>
<td></td>
<td>20.0</td>
<td></td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>29</td>
<td>97</td>
<td></td>
<td>97</td>
<td></td>
<td></td>
<td>20.0</td>
<td></td>
<td></td>
<td>96</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td></td>
<td></td>
<td>20</td>
<td>20</td>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>60</td>
<td></td>
<td></td>
<td>60</td>
<td>60</td>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td></td>
<td></td>
<td>20</td>
<td>20</td>
<td>20.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>60</td>
<td></td>
<td></td>
<td>60</td>
<td>60</td>
<td>20.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>NLN</th>
<th>ALN</th>
<th>DLN</th>
<th>IFAC</th>
<th>EFAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>10000.0</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>10000.0</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>10000.0</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>10000.0</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>10000.0</td>
<td></td>
<td></td>
<td>4</td>
<td>1000.0</td>
</tr>
<tr>
<td>39</td>
<td>10000.0</td>
<td></td>
<td></td>
<td>4</td>
<td>10.0</td>
</tr>
</tbody>
</table>
### TABLE 4 Continued

<table>
<thead>
<tr>
<th>Case</th>
<th>NLN</th>
<th>ALN</th>
<th>DLN</th>
<th>IFAC</th>
<th>EFAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>2</td>
<td>1000.0, 1000.0</td>
<td>1.0, 1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>3</td>
<td>1000.0, 1000.0, 1000.0</td>
<td>1.0, 1.0, 1.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** Blanks indicate the basic value (Case 1, see Table 3).
### TABLE 5

**Input Parameters for Basic Double-Body Field Used in Optimum Acceleration Parameter Studies**

**Case 42**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAX</td>
<td>93</td>
</tr>
<tr>
<td>JMAX</td>
<td>40</td>
</tr>
<tr>
<td>NBDY</td>
<td>2</td>
</tr>
<tr>
<td>IGES</td>
<td>0</td>
</tr>
<tr>
<td>NBSEG</td>
<td>4</td>
</tr>
<tr>
<td>NRSEG</td>
<td>2</td>
</tr>
<tr>
<td>LBSID</td>
<td>1 1 1 3</td>
</tr>
<tr>
<td>LB1</td>
<td>1 75 29 1</td>
</tr>
<tr>
<td>LB2</td>
<td>19 93 65 93</td>
</tr>
<tr>
<td>LBDY</td>
<td>1 1 2 0</td>
</tr>
<tr>
<td>LRSID</td>
<td>1 2</td>
</tr>
<tr>
<td>LR1</td>
<td>19 1</td>
</tr>
<tr>
<td>LR2</td>
<td>29 40</td>
</tr>
<tr>
<td>LISID</td>
<td>1 4</td>
</tr>
<tr>
<td>L11</td>
<td>65 1</td>
</tr>
<tr>
<td>L12</td>
<td>75 40</td>
</tr>
<tr>
<td>R(1)</td>
<td>varied</td>
</tr>
<tr>
<td>R(2)</td>
<td>0.0001</td>
</tr>
<tr>
<td>R(3)</td>
<td>0.0001</td>
</tr>
<tr>
<td>YINFIN</td>
<td>10.0</td>
</tr>
<tr>
<td>AINFIN</td>
<td>-30.0</td>
</tr>
<tr>
<td>XINF</td>
<td>0.0</td>
</tr>
<tr>
<td>YINF</td>
<td>0.0</td>
</tr>
<tr>
<td>NINF</td>
<td>92</td>
</tr>
<tr>
<td>IEV</td>
<td>0</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>R(10)</td>
<td>0.0</td>
</tr>
<tr>
<td>INFAC</td>
<td>0</td>
</tr>
<tr>
<td>INFACO</td>
<td>0</td>
</tr>
<tr>
<td>ATYP</td>
<td>ETA</td>
</tr>
<tr>
<td>ITYP</td>
<td>0</td>
</tr>
<tr>
<td>NLN</td>
<td>1</td>
</tr>
<tr>
<td>NPT</td>
<td>0</td>
</tr>
<tr>
<td>DEC</td>
<td>0.0</td>
</tr>
<tr>
<td>AMPFAC</td>
<td>0.0</td>
</tr>
<tr>
<td>JLN</td>
<td>1</td>
</tr>
<tr>
<td>ALN</td>
<td>1000.0</td>
</tr>
<tr>
<td>DLN</td>
<td>1.0</td>
</tr>
<tr>
<td>ATYP</td>
<td>0</td>
</tr>
<tr>
<td>ITYP</td>
<td>0</td>
</tr>
<tr>
<td>NLN</td>
<td>0</td>
</tr>
<tr>
<td>NPT</td>
<td>0</td>
</tr>
<tr>
<td>DEC</td>
<td>0.0</td>
</tr>
<tr>
<td>AMPFAC</td>
<td>0.0</td>
</tr>
<tr>
<td>IFAC</td>
<td>3</td>
</tr>
<tr>
<td>EFAC</td>
<td>100.0</td>
</tr>
</tbody>
</table>
TABLE 6
Input Parameters Varied for Double-Body Field Used in Optimum Acceleration Parameter Studies

<table>
<thead>
<tr>
<th>Case</th>
<th>IMAX</th>
<th>JMAX</th>
<th>LB1</th>
<th>LB2</th>
<th>LR1</th>
<th>LR2</th>
<th>LI1</th>
<th>LI2</th>
<th>YINFIN</th>
<th>NINF</th>
<th>ALN</th>
<th>IFAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>60</td>
<td></td>
<td>19</td>
<td>1</td>
<td>29</td>
<td>60</td>
<td>65</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>20</td>
<td></td>
<td>19</td>
<td>1</td>
<td>29</td>
<td>60</td>
<td>65</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10000.0</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>149</td>
<td></td>
<td>1</td>
<td>117</td>
<td>33</td>
<td>149</td>
<td>33</td>
<td>149</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>43</td>
<td>40</td>
<td>107</td>
<td>117</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>40</td>
<td>1</td>
<td>40</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>149</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Blanks indicate the basic value (Case 42, see Table 5).
<table>
<thead>
<tr>
<th>Case Number</th>
<th>Case</th>
<th>Experimental Optimum</th>
<th>Variable Field</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Optimum Acceleration Parameter</td>
<td>Number of Iterations</td>
</tr>
<tr>
<td>Effect of Number of Points on Airfoil</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>IMAX = 37 (IMAX Down)</td>
<td>1.77</td>
<td>68</td>
</tr>
<tr>
<td>1</td>
<td>IMAX = 67 (IMAX Basic)</td>
<td>1.84</td>
<td>74</td>
</tr>
<tr>
<td>5</td>
<td>IMAX = 97 (IMAX Up)</td>
<td>1.88</td>
<td>137</td>
</tr>
<tr>
<td>Effect of JMAX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>JMAX = 20 (JMAX Down)</td>
<td>1.84</td>
<td>90</td>
</tr>
<tr>
<td>1</td>
<td>JMAX = 40 (JMAX Basic)</td>
<td>1.84</td>
<td>74</td>
</tr>
<tr>
<td>7</td>
<td>JMAX = 60 (JMAX Up)</td>
<td>1.85</td>
<td>126</td>
</tr>
<tr>
<td>Effect of Attraction Amplitude</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Amplitude = 100 (Amplitude Down)</td>
<td>1.85</td>
<td>81</td>
</tr>
<tr>
<td>1</td>
<td>Amplitude = 1000 (Amplitude Basic)</td>
<td>1.84</td>
<td>74</td>
</tr>
<tr>
<td>3</td>
<td>Amplitude = 10,000 (Amplitude Up)</td>
<td>1.86</td>
<td>114</td>
</tr>
<tr>
<td>Effect of Outer Boundary Radius</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Radius = 5 (Radius Down)</td>
<td>1.80</td>
<td>205</td>
</tr>
<tr>
<td>1</td>
<td>Radius = 10 (Radius Basic)</td>
<td>1.84</td>
<td>74</td>
</tr>
<tr>
<td>9</td>
<td>Radius = 20 (Radius Up)</td>
<td>1.84</td>
<td>184</td>
</tr>
<tr>
<td>Effect of Body Shape</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Airfoil</td>
<td>1.84</td>
<td>74</td>
</tr>
<tr>
<td>0</td>
<td>Circle</td>
<td>1.88</td>
<td>96</td>
</tr>
</tbody>
</table>
TABLE 8
Optimum Acceleration Parameters for Single-Body
Field : Variation of Pairs of Quantities

<table>
<thead>
<tr>
<th>(12) IMAX Up</th>
<th>(5) IMAX Up</th>
<th>(13) IMAX Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplitude Down</td>
<td>1.89 : 122</td>
<td>1.88 : 137</td>
</tr>
<tr>
<td></td>
<td>(1.90 : &gt;122)</td>
<td>(1.89 : &gt;183)</td>
</tr>
<tr>
<td>(2) Amplitude Down</td>
<td></td>
<td>(3) Amplitude Up</td>
</tr>
<tr>
<td>1.85 : 81</td>
<td></td>
<td>1.86 : 114</td>
</tr>
<tr>
<td></td>
<td>(4) IMAX Down</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.79 : 63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1.85 : 92)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10) IMAX Down</td>
<td></td>
</tr>
<tr>
<td>Amplitude Down</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.84 : 74</td>
<td>(1.84 : 83)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) JMAX Down</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.84 : 90</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(17) JMAX Up</td>
<td></td>
</tr>
<tr>
<td>Amplitude Down</td>
<td>1.86 : 129</td>
<td>1.87 : 147</td>
</tr>
<tr>
<td></td>
<td>(1.90 : DIV)</td>
<td>(1.90 : DIV)</td>
</tr>
<tr>
<td>(2) Amplitude Down</td>
<td></td>
<td>(3) Amplitude Up</td>
</tr>
<tr>
<td>1.85 : 81</td>
<td>1.84 : 74</td>
<td>1.86 : 114</td>
</tr>
<tr>
<td></td>
<td>(15) JMAX Down</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.84 : 94</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(14) JMAX Down</td>
<td></td>
</tr>
<tr>
<td>Amplitude Down</td>
<td>1.84 : 89</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1.84 : 94)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(9) Radius Up</td>
<td></td>
</tr>
<tr>
<td>Amplitude Down</td>
<td>1.89 : 131</td>
<td>1.84 : 205</td>
</tr>
<tr>
<td></td>
<td>(1.88 : &gt;131)</td>
<td>(1.88 : DIV)</td>
</tr>
<tr>
<td>(2) Amplitude Down</td>
<td></td>
<td>(3) Amplitude Up</td>
</tr>
<tr>
<td>1.85 : 81</td>
<td>1.84 : 74</td>
<td>1.86 : 114</td>
</tr>
<tr>
<td></td>
<td>(18) Radius Down</td>
<td></td>
</tr>
<tr>
<td>Amplitude Down</td>
<td>1.80 : 185</td>
<td>1.80 : 184</td>
</tr>
<tr>
<td></td>
<td>(1.91 : DIV)</td>
<td></td>
</tr>
<tr>
<td>(20) Radius Up</td>
<td>(19) Radius Down</td>
<td></td>
</tr>
<tr>
<td>Amplitude Down</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.89 : 131</td>
<td>1.81 : 176</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1.88 : &gt;131)</td>
<td>(1.91 : DIV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>(24) JMAX Up</td>
<td>( 7) JMAX Up</td>
<td>(25) JMAX Up</td>
</tr>
<tr>
<td>IMAX Down</td>
<td></td>
<td>IMAX Up</td>
</tr>
<tr>
<td>1.83 : 77</td>
<td>1.85 : 126</td>
<td>1.89 : 128</td>
</tr>
<tr>
<td>(1.89 : &gt;100)</td>
<td></td>
<td>(1.91 : DIV)</td>
</tr>
<tr>
<td>( 4) IMAX Down</td>
<td>( 1) Basic</td>
<td>( 5) IMAX Up</td>
</tr>
<tr>
<td></td>
<td>1.84 : 74</td>
<td>1.88 : 137</td>
</tr>
<tr>
<td>(22) JMAX Down</td>
<td>( 6) JMAX Down</td>
<td>(23) JMAX Down</td>
</tr>
<tr>
<td>IMAX Down</td>
<td></td>
<td>IMAX Up</td>
</tr>
<tr>
<td>1.71 : 48</td>
<td>1.81 : 90</td>
<td>1.84 : 142</td>
</tr>
<tr>
<td>(1.76 : 62)</td>
<td></td>
<td>(1.87 : &gt;142)</td>
</tr>
<tr>
<td>(28) Radius Up</td>
<td>( 9) Radius Up</td>
<td>(29) Radius Up</td>
</tr>
<tr>
<td>IMAX Down</td>
<td></td>
<td>IMAX Up</td>
</tr>
<tr>
<td>1.81 : 77</td>
<td>1.84 : 205</td>
<td>1.87 : 219</td>
</tr>
<tr>
<td>(1.84 : 92)</td>
<td></td>
<td>(1.90 : &gt;219)</td>
</tr>
<tr>
<td>( 4) IMAX Down</td>
<td>( 1) Basic</td>
<td>( 5) IMAX Up</td>
</tr>
<tr>
<td></td>
<td>1.84 : 74</td>
<td>1.88 : 137</td>
</tr>
<tr>
<td>(26) Radius Down</td>
<td>( 8) Radius Down</td>
<td>(27) Radius Down</td>
</tr>
<tr>
<td>IMAX Down</td>
<td></td>
<td>IMAX Up</td>
</tr>
<tr>
<td>1.78 : 102</td>
<td>1.80 : 184</td>
<td>1.82 : 285</td>
</tr>
<tr>
<td>(1.84 : &gt;102)</td>
<td></td>
<td>(1.93 : DIV)</td>
</tr>
<tr>
<td>(32) Radius Up</td>
<td>( 9) Radius Up</td>
<td>(33) Radius Up</td>
</tr>
<tr>
<td>JMAX Down</td>
<td></td>
<td>JMAX Up</td>
</tr>
<tr>
<td>1.82 : 117</td>
<td>1.84 : 205</td>
<td>1.89 : 103</td>
</tr>
<tr>
<td>(1.84 : &gt;117)</td>
<td></td>
<td>(1.90 : &gt;103)</td>
</tr>
<tr>
<td>( 6) JMAX Down</td>
<td>( 1) Basic</td>
<td>( 7) JMAX Up</td>
</tr>
<tr>
<td></td>
<td>1.84 : 74</td>
<td>1.85 : 126</td>
</tr>
<tr>
<td>(30) Radius Down</td>
<td>( 8) Radius Down</td>
<td>(31) Radius Down</td>
</tr>
<tr>
<td>JMAX Down</td>
<td></td>
<td>JMAX Up</td>
</tr>
<tr>
<td>1.80 : 176</td>
<td>1.80 : 184</td>
<td>1.58 : 218</td>
</tr>
<tr>
<td>(1.91 : DIV)</td>
<td></td>
<td>(1.91 : DIV)</td>
</tr>
</tbody>
</table>
TABLE 8 Continued

Format: (Case Number) Case
Optimum Acceleration Parameter: Number of Iterations
(Average Variable Acceleration Parameter: Number of Iterations)

See Tables 3-4 for values of the quantities varied.
**TABLE 9**

Optimum Acceleration Parameter
for Single-Body Field: Miscellaneous Variations

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Case</th>
<th>Experimental Optimum</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Optimum Acceleration Parameter</strong></td>
<td><strong>Number of Iterations</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>One Step</td>
<td>1.35</td>
<td>561</td>
</tr>
<tr>
<td>34</td>
<td>Two Step</td>
<td>1.61</td>
<td>317</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Three Step</td>
<td>1.85</td>
<td>119</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Four Step</td>
<td>1.86</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Five Step</td>
<td>1.85</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Effect of Number of Steps in Addition of Inhomogeneous Term (Attraction Up)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>One Step</td>
<td>1.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Two Step</td>
<td>1.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Three Step</td>
<td>1.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Four Step</td>
<td>1.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Five Step</td>
<td>1.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Effect of Intermediate Convergence Criterion (Attraction Up)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Convergence : -01</td>
<td>1.86</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Convergence : -02</td>
<td>1.86</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Convergence : -03</td>
<td>1.85</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Effect of Number of Attraction Lines (Basic)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>One Line</td>
<td>1.84</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Two Lines</td>
<td>1.83</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Three Lines</td>
<td>1.83</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Effect of Convergence Criterion (Basic)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Convergence : -03</td>
<td>1.84</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Convergence : -04</td>
<td>1.84</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Convergence : -05</td>
<td>1.84</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Convergence : -06</td>
<td>1.89</td>
<td>175</td>
<td></td>
</tr>
</tbody>
</table>
## TABLE 10

Optimum Acceleration Parameters for Two-Body Field

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Case</th>
<th>Optimum Acceleration Parameter</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>Effect of Number of Points on Airfoils</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>37 Points (IMAX Basic)</td>
<td>1.84</td>
<td>189</td>
</tr>
<tr>
<td></td>
<td>65 Points (IMAX Up)</td>
<td>1.86</td>
<td>304</td>
</tr>
<tr>
<td>44</td>
<td>Effect of JMAX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JMAX = 20 (.JMAX Down)</td>
<td>1.78</td>
<td>203</td>
</tr>
<tr>
<td></td>
<td>JMAX = 40 (.JMAX Basic)</td>
<td>1.84</td>
<td>189</td>
</tr>
<tr>
<td></td>
<td>JMAX = 60 (.JMAX Up)</td>
<td>1.85</td>
<td>238</td>
</tr>
<tr>
<td>46</td>
<td>Effect of Attraction Amplitude</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amplitude = 100 (Amplitude Down)</td>
<td>1.85</td>
<td>176</td>
</tr>
<tr>
<td></td>
<td>Amplitude = 1000 (Amplitude Basic)</td>
<td>1.84</td>
<td>189</td>
</tr>
<tr>
<td></td>
<td>Amplitude = 10,000 (Amplitude Up)</td>
<td>1.59</td>
<td>477</td>
</tr>
<tr>
<td>48</td>
<td>Effect of Outer Boundary Radius</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Radius = 5 (Radius Down)</td>
<td>1.61</td>
<td>285</td>
</tr>
<tr>
<td></td>
<td>Radius = 10 (Radius Basic)</td>
<td>1.84</td>
<td>189</td>
</tr>
<tr>
<td></td>
<td>Radius = 20 (Radius Up)</td>
<td>1.86</td>
<td>263</td>
</tr>
<tr>
<td>50</td>
<td>Effect of Number of Steps in Addition of Inhomogeneous Term (Attraction Up)</td>
<td>Divergence</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>Three Steps</td>
<td>1.59</td>
<td>466</td>
</tr>
<tr>
<td>52</td>
<td>Four Steps</td>
<td>1.59</td>
<td>472</td>
</tr>
<tr>
<td>47</td>
<td>Six Steps</td>
<td>1.59</td>
<td>477</td>
</tr>
</tbody>
</table>
### TABLE 10 Continued

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Case</th>
<th>Optimum Acceleration Parameter</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(52)</td>
<td>Effect of Intermediate Convergence Criterion (Attraction Up)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(53)</td>
<td>Five Steps, Convergence : -02</td>
<td>1.59</td>
<td>472</td>
</tr>
<tr>
<td>(47)</td>
<td>Five Steps, Convergence : -03</td>
<td>1.63</td>
<td>570</td>
</tr>
<tr>
<td>(54)</td>
<td>Six Steps, Convergence : -02</td>
<td>1.59</td>
<td>477</td>
</tr>
<tr>
<td></td>
<td>Six Steps, Convergence : -03</td>
<td>1.63</td>
<td>602</td>
</tr>
</tbody>
</table>
TABLE II

Input Parameters for Coordinate System Control Demonstration of Figure 27

\[ IMAX = 31, JMAX = 30, YINFIN = 10, AINFIN = 0.0 \]

(a.) No Attraction

(b.) ETA Attraction: \( JLN = 20, ALN = 10.0, DLN = 1.0 \)

(c.) ETA Attraction: \( IPT = 1, JPT = 20, APT = 10.0, DPT = 1.0 \)
    \( IPT = 16, JPT = 20, APT = 10.0, DPT = 1.0 \)

(d.) ETA Attraction: \( IPT = 5, JPT = 2, APT = 1000.0, DPT = 1.0 \)
    \( IPT = 6, JPT = 2, APT = 1000.0, DPT = 100.0 \)

(e.) XI Attraction: \( JLN = 2, ALN = 100.0, DLN = 1.0 \)

(f.) XI Attraction: \( IPT = 5, JPT = 1, APT = 1000.0, DPT = 1.0 \)
Table 12. Coordinate System Control Parameters for Figures 33, 34, 37, and 38.

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 33.</td>
<td>Coordinate line attraction to the first 15 lines around the airfoil with amplitude 20,000 on the body, varying linearly to 13,000 on the 15th line. Decay factor of 1.0 for all except last line, where 0.4 was used. 75 points on airfoil, 58 lines around airfoil. Circular boundary of radius 10 chords.</td>
</tr>
<tr>
<td>Figure 34.</td>
<td>Coordinate line attraction to the first 15 lines around the airfoil with amplitude 20,000 on the body, varying linearly to 13,000 on the 15th line. Decay factor of 1.0 for all except last line, where 0.4 was used. 75 points on airfoil, 58 lines around airfoil. Circular boundary of radius 10 chords.</td>
</tr>
<tr>
<td>Figure 37.</td>
<td>Coordinate line attraction to the entire boundary with amplitude 100 and decay factor 0.1.</td>
</tr>
<tr>
<td>Figure 38.</td>
<td>Coordinate line attraction to the first 15 lines around the body with amplitude 20,000 on the body varying linearly to 13,000 on the 15th line. Decay factor of 1.0 for all except last line where 0.4 was used. 75 points on body, 58 lines around body. Circular boundary of radius 10 chords.</td>
</tr>
</tbody>
</table>
Figure 1. Field Transformation - Single Body
Figure 2. Field Transformation - Multiple Bodies
Figure 3. Computational Grids - Single and Two Body Regions
Figure 4a. Single-Body Configuration
Figure 4b. Alternate Single-Body Configuration
Figure 5. Double-Body Segment Configuration #1
Figure 6. Double-Body Segment Configuration #2
Figure 7. Double-Body Segment Configuration #3
Figure 8. Double-Body Segment Configuration #4
Figure 9. Double-Body Segment Configuration #5
Figure 10. Double-Body Segment Configuration #6
Figure 11. Double-Body Segment Configuration #7
Figure 12. Double-Body Segment Configuration #8
Figure 13. Double-Body Segment Configuration #9
Figure 14. Initial Guess Types - Single Body
Figure 15. Initial Guess Types - Double-Body Segment Configuration #1
Figure 16. Initial Guess Types - Double-Body Segment Configuration #2
Figure 17. Initial Guess Types - Double-Body Segment Configuration #3
Figure 18. Initial Guess Types – Double-Body Segment Configuration #4
Figure 19. Initial Guess Types - Double-Body Segment Configuration #5
Figure 20. Initial Guess Types - Double-Body Segment Configuration #6
Figure 21. Initial Guess Types - Double-Body Segment Configuration #7
Figure 22. Initial Guess Types - Double-Body Segment Configuration #8
Figure 23. Initial Guess Types - Double-Body Segment Configuration #9
Figure 26. Effect of Acceleration Parameter on Convergence Rate
Figure 27. Examples of Coordinate System Control

(See Table 11 for control parameters.)
Figure 28. Example of Attraction into Concave Region
Figure 29. Coordinate Systems - Circular Cylinder And Joukowski Airfoil
Figure 30. Coordinate Systems - Karman-Trefftz Airfoils, No Coordinate Line Contraction
Figure 31. Coordinate Systems - Karman-Trefftz Airfoils, With Coordinate Line Contraction
Figure 33. Contracted Coordinate System - Göttingen 625 Airfoil (See Table 12.)
Figure 34. Contracted Coordinate System - NACA 0018 Airfoil (See Table 12.)
Figure 35. Contracted Coordinate System - Multiple Airfoil

Coordinate line attraction to the first 10 lines around the airfoils with amplitude 10,000 on the body. Decay factor of 1.0 for all except last line, where 0.5 was used. 37 points on airfoils, 40 lines around airfoil. Circular boundary of radius 10 fore body chords.
Figure 36. Coordinate System - Triangular Simply-Connected Region.
Figure 37. Coordinate System - Key-seat Shaft, Simply-Connected Region
(See Table 12.)
Figure 39. Comparison of Numerical and Analytic Potential Flow Solutions for Flapped Karman-Trefftz Airfoil. No coordinate attraction.
Figure 40. Comparison of Experimental and Numerical Pressure Distribution - Liebeck Laminar Airfoil
Figure 41. Potential Flow Streamlines and Pressure Distributions
- Double Airfoil
Figure 42. Coordinate System - Double Cylinders, Coordinate Attraction to Cut Intersections

Attraction to intersections of body contours with cut between at amplitude 1000 and decay factor 0.5. 31 points on each obdy. 40 lines around the bodies. Outer boundary at 20 diameters.
Figure 43. Potential Flow Streamlines and Pressure Distribution - Double Cylinder
Figure 44. Potential Flow Pressure Distribution for Double Cylinder without Coordinate Attraction
Figure 45. Alternate Segment Arrangement - Double Cylinders
No coordinate attraction.
Figure 46. Potential Flow Pressure Distribution for Double Cylinders - Alternate Segment Arrangement
Figure 47. Streamlines - Göttingen 625 Airfoil, $R = 2000$, $\alpha = 5^\circ$
Figure 48. Velocity Profiles, Gottingen 625 Airfoil, $R = 2000$, $\alpha = 5^\circ$, $t = 1.83$. 
Figure 49. Pressure Distribution - Göttingen 625 Airfoil, 
$R = 2000$, $\alpha = 5^\circ$, $t = 0.118$ and 3.33
Figure 50. Streamlines and Vorticity Contours - Rock, $R = 500$, $t = 0.5$
Figure 51. Velocity Profiles - Rock, $R = 500$, $t = 1.2$
Figure 52a. Pressure Distribution and Velocity Vectors - Double Airfoil, $R = 1000$, $t = 0.1$
Figure 52b. Pressure Distribution and Velocity Vectors - Double Airfoil, $R = 1000$, $t = 1.2$
Figure 52c. Pressure Distribution and Velocity Vectors - Double Airfoil, R = 1000, t = 3.02
Fig. 53. Comparison of Numerical and Analytic Solution for Deflection Contours for a Simply-Supported Uniformly Loaded Plate.
APPENDIX A

DERIVATIVES AND VECTORS IN THE TRANSFORMED PLANE

This appendix contains a comprehensive set of relations in the transformed \([\xi, \eta]\) plane. A few relations involving \(x\) and \(y\) derivatives of the coordinate functions \(\xi(x,y)\) and \(\eta(x,y)\) are also included. Since the intent here is to provide a quick reference only, most of the algebraic development is omitted. The following function definitions are applicable throughout this appendix:

\[
f(x,y,t) - a twice continuously differentiable scalar function of \(x, y,\) and \(t.
\]

\[
F(x,y) = i F_1(x,y) + j F_2(x,y) - a continuously differentiable vector-valued function of \(x\) and \(y.\) \(i\) and \(j\) are the conventional cartesian coordinate unit vectors.
\]

\[
J = x_\xi y_\eta - x_\eta y_\xi
\]

\[
\alpha = x_\eta^2 + y_\eta^2
\]

\[
\beta = x_\xi x_\eta + y_\xi y_\eta
\]

\[
\gamma = x_\xi^2 + y_\xi^2
\]

\[
D_x = \alpha x_\xi^{\xi} - 2\beta x_\xi^{\eta} + \gamma x_\eta^{\eta}
\]

\[
D_y = \alpha y_\xi^{\xi} - 2\beta y_\xi^{\eta} + \gamma y_\eta^{\eta}
\]

\[
\sigma = (y_\xi D_x - x_\xi D_y)/J
\]

\[
\tau = (x_\eta D_y - y_\eta D_x)/J
\]
Derivative Transformations

\[ f_x = \frac{\partial f}{\partial x} x, y, t = (y f_x - y f_y) / J \]  
(A.1)

\[ f_y = \frac{\partial f}{\partial y} x, y, t = (x f_x - x f_y) / J \]  
(A.2)

\[ f_t = \frac{\partial f}{\partial t} x, y, t = \left( \frac{\partial f}{\partial t} \right)_{x, y, t} - \frac{1}{J} \left( y f_x - y f_y \right) \left( \frac{\partial x}{\partial t} \right)_{x, y, t} - \frac{1}{J} \left( y f_x - x f_y \right) \left( \frac{\partial y}{\partial t} \right)_{x, y, t} \]  
(A.3)

\[ f_{xx} = \frac{\partial^2 f}{\partial x^2} x, y, t = (y f_{xx} + 2y f_x f_{x,y} + y f_{yy}) / J^2 \] 
\[ + \left[ (y f_{xx} + 2y f_x f_{x,y} + y f_{yy}) \left( f_{x,y} - f_{y} \right) \right] \] 
\[ + \left[ (y f_{xx} + 2y f_x f_{x,y} + y f_{yy}) \left( f_{x,y} - f_{y} \right) \right] / J^3 \]  
(A.4)

\[ f_{yy} = \frac{\partial^2 f}{\partial y^2} x, y, t = (x f_{yy} + 2x f_y f_{x,y} + x f_{xx}) / J^2 \] 
\[ + \left[ (x f_{yy} + 2x f_y f_{x,y} + x f_{xx}) \left( f_{x,y} - f_{y} \right) \right] \] 
\[ + \left[ (x f_{yy} + 2x f_y f_{x,y} + x f_{xx}) \left( f_{x,y} - f_{y} \right) \right] / J^3 \]  
(A.5)

\[ f_{xy} = \left[ (x f_{x,y} + x f_{y,x}) f_{x,y} - x f_{x,y} f_{x,y} - x f_{y,x} f_{x,y} \right] / J^2 \] 
\[ + \left[ (x f_{x,y} + x f_{y,x}) f_{x,y} - x f_{x,y} f_{x,y} - x f_{y,x} f_{x,y} \right] \] 
\[ + \left[ (x f_{x,y} + x f_{y,x}) f_{x,y} - x f_{x,y} f_{x,y} - x f_{y,x} f_{x,y} \right] / J^3 \]  
(A.6)
Derivatives of $\xi(x,y)$ and $n(x,y)$

$$\xi_x = y_n/J \quad \text{(A.7)}$$
$$\xi_y = -x_n/J \quad \text{(A.8)}$$
$$n_x = -y_\xi/J \quad \text{(A.9)}$$
$$n_y = x_\xi/J \quad \text{(A.10)}$$

$$\xi_{xx} = (\xi_x y_\xi + n_x y_n)/J - (\xi_x^2 J_\xi + \xi_x n_x J_n)/J \quad \text{(A.11)}$$
$$\xi_{yy} = - (n_y x_n + \xi_y x_\xi)/J - (\xi_y n_y J_\xi + \xi_y^2 J_\xi)/J \quad \text{(A.12)}$$

$$\xi_{xy} = (n_y y_n + \xi_y y_\xi n)/J - (\xi_x y_\xi J_\xi + \xi_x n_x J_n)/J \quad \text{(A.13)}$$
$$n_{xx} = - (\xi_x y_\xi + n_x y_\xi)/J - (\xi_x n_x J_\xi + n_x^2 J_n)/J \quad \text{(A.14)}$$
$$n_{yy} = (n_y x_\xi n + \xi_y x_\xi)/J - (\xi_y n_y J_\xi + n_y^2 J_\xi)/J \quad \text{(A.15)}$$
$$n_{xy} = - (n_y y_\xi n + \xi_y y_\xi)/J - (n_x n_y J_n + \xi_y n_x J_\xi)/J \quad \text{(A.16)}$$
Vector Derivative Transformations

Laplacian:

\[
\nabla^2 f = \frac{(a f_{\xi\xi} - 2\beta f_{\xi\eta} + \gamma f_{\eta\eta})}{J^2} + \left[ (ax_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta})(y f_{\eta\eta} - y f_{\xi\xi})\right]
\]

or

\[
\nabla^2 f = \frac{(a f_{\xi\xi} - 2\beta f_{\xi\eta} + \gamma f_{\eta\eta} + \sigma f_{\eta\eta} + \tau f_{\xi\eta})}{J^2}
\]

Gradient:

\[
\nabla f = \frac{[(y f_{\eta\xi} - y f_{\xi\eta})x + (x f_{\eta\xi} - x f_{\xi\eta})y]}{J}
\]

Divergence:

\[
\nabla \cdot F = \frac{[y_n(F_{1\xi}) - y_x(F_{1\eta}) + x_n(F_{2\eta}) - x_x(F_{2\xi})]}{J}
\]

Curl:

\[
\nabla \times F = \frac{k[y_n(F_{2\xi}) - y_x(F_{2\eta}) - x_n(F_{1\eta}) + x_x(F_{1\xi})]}{J}
\]
Unit Tangent and Unit Normal Vectors in the \( \xi, \eta \) Plane

In many applications components of vector valued functions either normal or tangent to a line of constant \( \xi \) or \( \eta \) are required. Similarly, directional derivatives in these directions are often needed to evaluate boundary conditions. These quantities may be obtained by trivial calculations if unit vectors tangent and normal to the \( \xi \) and \( \eta \)-lines are available. These vectors are developed below.

It is well known that the unit normal to the graph of \( f(x,y) = \) constant is given by

\[
\hat{n}(f) = \frac{\nabla f}{|\nabla f|}
\]

Associating the coordinate function \( \eta(x,y) \) with \( f(x,y) \), we have

\[
\hat{n}(\eta) = \frac{\nabla \eta}{|\nabla \eta|}
\]

Utilizing equation (A.19) this reduces to

\[
\hat{n}(\eta) = \frac{y \hat{i} - x \hat{j}}{\sqrt{\gamma}}
\]

which is the unit vector normal to a line of constant \( \eta \). In a similar manner the unit vector normal to a line of constant \( \xi \) is given by

\[
\hat{n}(\xi) = \frac{\nabla \xi}{|\nabla \xi|} = \frac{y \hat{i} - x \hat{j}}{\sqrt{\gamma}}
\]

These vectors are illustrated as they appear in the physical plane
in Figure A.1 below.

![Figure A.1. Unit Tangent and Normal Vectors](image)

The unit tangent vectors are then given by

\[ t(\eta) = \hat{n}(\eta) \times \hat{k} = \left( x_\eta \hat{i} + y_\eta \hat{j} \right) / \sqrt{\gamma} \quad (A.24) \]

\[ t(\xi) = \hat{n}(\xi) \times \hat{k} = -\left( x_\xi \hat{i} + y_\xi \hat{j} \right) / \sqrt{\alpha} \quad (A.25) \]

**Vector Components Tangent and Normal to Lines of Constant \( \xi \) and \( \eta \)**

\[ F_\eta(\eta) = \hat{n}(\eta) \cdot \hat{F} = \left( -y_\eta F_1 + x_\eta F_2 \right) / \sqrt{\gamma} \quad (A.26) \]

\[ F_\xi(\eta) = \hat{t}(\eta) \cdot \hat{F} = \left( x_\xi F_1 + y_\xi F_2 \right) / \sqrt{\gamma} \quad (A.27) \]
\[ F_\xi (\xi) = n(\xi) \cdot F = (y_\eta F_1 - x_\eta F_2)/\sqrt{\alpha} \quad (A.28) \]

\[ F_t (\xi) = t(\xi) \cdot F = - (x_\eta F_1 + y_\eta F_2)/\sqrt{\alpha} \quad (A.29) \]

**Directional Derivatives**

\[ \frac{\partial f}{\partial n}(\eta) = n(\eta) \cdot \nabla f = (\gamma f + \beta f_\xi)/\sqrt{\gamma} \quad (A.30) \]

\[ \frac{\partial f}{\partial t}(\eta) = t(\eta) \cdot \nabla f = f_\xi/\sqrt{\gamma} \quad (A.31) \]

\[ \frac{\partial f}{\partial n}(\xi) = n(\xi) \cdot \nabla f = (\alpha f_\xi + \beta f_\eta)/\sqrt{\alpha} \quad (A.32) \]

\[ \frac{\partial f}{\partial t}(\xi) = t(\xi) \cdot \nabla f = - f_\eta/\sqrt{\alpha} \quad (A.33) \]

**Integral Transform**

**Scalar Function:**

\[ \int f(x,y) \, dx \, dy = \int_{\mathbb{D}} f(x(\xi,\eta), y(\xi,\eta)) \, d\xi \, d\eta \quad (A.34) \]

**Vector Function:**

Consider a vector integral in the physical plane of the form

\[ I = \int_S f(x,y) \, n(x,y) \, dS \quad (A.35) \]

where \( S \) is the closed cylindrical surface of unit depth whose perimeter is specified by the contour \( \Gamma_1 \) in the physical plane (see Figure A.2) and whose outward unit normal at any point is given by \( n(x,y) \).
Figure A.2. Integration Around Contour $\Gamma_1$

If $r = r(x,y)$ is the position vector describing $\Gamma_1$, then

$$dS = (1.0) \ |dr_\perp|$$

which implies that (A.35) becomes

$$I = \oint \phi f(x,y) n(x,y) \ |dr_\perp|$$

We now wish to transform (A.36) to the $(\xi, \eta)$ plane. Consider $|dr_\perp|:

$$|dr_\perp| = \sqrt{(dx)^2 + (dy)^2} = \sqrt{\gamma (d\xi)^2 + \beta d\xi d\eta + \alpha (d\eta)^2}$$

$$= \sqrt{\gamma} \ d\xi$$

(A.37)

since $\Gamma_1$ transforms to $\Gamma_1^*$, a constant $\eta$-line ($\eta = \eta_1$). Noting that

$$n(x,y) = \eta_1 = (-y_{\xi_1} + x_{\xi_1})/\sqrt{\gamma} \ (\text{Equation (A.22)})$$

Equation (A.37) becomes
\[ I = \int_{\xi_{\text{min}}}^{\xi_{\text{max}}} f(x(\xi, \eta_1), y(\xi, \eta_1)) (x_{\xi} - y_{\xi}) d\xi \]

\[ = \int_{\xi_{\text{min}}}^{\xi_{\text{max}}} f(\xi, \eta_1) (x_{\xi} - y_{\xi}) d\xi \]

(A.38)

where \( \xi_{\text{min}} \) and \( \xi_{\text{max}} \) are the minimum and maximum values respectively of \( \xi \) on \( \Gamma \). Note that all quantities in (A.38) are evaluated along \( \eta = \eta_1 \). If the vector \( n(x, y) \) is incorporated into the function \( f(x, y) \), we can define the vector function \( f(x, y) \) as

\[ f(x, y) \equiv f(x, y) n(x, y) \]

Equation (A.38) now becomes

\[ I = \int_{\xi_{\text{min}}}^{\xi_{\text{max}}} f(\xi, \eta_1) \sqrt{\gamma} d\xi \]

(A.39)

which is merely an alternate form of (A.38).
APPENDIX B

PRESERVATION OF EQUATION TYPE

When coordinate transformations are utilized as an aid in solving a given partial differential equation, it is imperative that the equation not change type under the transformation. Such an invariance will now be demonstrated for the transformation discussed in Section II. Consider the general, second order, quasi-linear partial differential equation

\[ A(x,y,f)f_{xx} + B(x,y,f)f_{xy} + C(x,y,f)f_{yy} + E(x,y,f)f_x + F(x,y,f)f_y + G(x,y,f) = 0 \]  

where \( f = f(x,y) \) is a twice continuously differentiable scalar function and \( A, B, C, E, F, \) and \( G \) are continuous functions. Recall that the equation type is determined by the coefficient functions \( A, B, \) and \( C \) as follows:

Elliptic if \( B^2 - 4AC < 0 \)

Parabolic if \( B^2 - 4AC = 0 \)

Hyperbolic if \( B^2 - 4AC > 0 \)

Utilizing equations (A.1), (A.2), (A.4) - (A.6), and (A.7) - (A.10) of Appendix A, equation (B.1) transforms to

\[ A*f_{\xi \xi} + B*f_{\xi \eta} + C*f_{\eta \eta} + E*f_{\xi} + F*f_{\eta} + G* = 0 \]  

(B.2)
where:

\[ A^* \equiv A\xi_x^2 + B\xi_x\xi_y + C\xi_y^2 \]

\[ B^* \equiv 2A\xi_x\eta_x + B(\xi_x\eta_y + \xi_y\eta_x) + 2C\xi_y\eta_y \]

\[ C^* \equiv A\eta_x^2 + B\eta_x\eta_y + C\eta_y^2 \]

\[ E^* \equiv A\xi_{xx} + B\xi_{xy} + C\xi_{yy} + E\xi_x + F\xi_y \]

\[ F^* \equiv A\eta_{xx} + B\eta_{xy} + C\eta_{yy} + E\eta_x + F\eta_y \]

\[ F^* \equiv F \]

Now, consider \((B^*)^2 - 4A^*C^*:\)

\[
(B^*)^2 - 4A^*C^* = [2A\xi_x\eta_x + B(\xi_x\eta_y + \eta_x\xi_y) + 2C\xi_y\eta_y]^2
\]

\[ - 4(A\xi_x^2 + B\xi_x\xi_y + C\xi_y^2)(A\eta_x^2 + B\eta_x\eta_y + C\eta_y^2) \]

\[ = (B^2 - 4AC)(\xi_x\eta_y - \xi_y\eta_x)^2 \]

\[ = (B^2 - 4AC)/J^2 \]

Since \(J^2 > 0\), \(B^2 - 4AC\) and \((B^*)^2 - 4A^*C^*\) are either both positive, both negative, or both zero. This implies that (B.1) and (B.2) have the same type.
APPENDIX C

PRESERVATION OF INTEGRAL CONSERVATION RELATIONS
IN THE TRANSFORMED PLANE

It is now shown that the divergence property is not lost in the transformed plane. Let \( \mathbf{F}(x,y) \) be a vector valued function defined on the physical plane (Region D in Figure 1) whose component functions, \( F_1(x,y) \) and \( F_2(x,y) \), are continuously differentiable on D. Let \( S(x,y) \) be a continuously differentiable function on D and suppose \( F(x,y) \) and \( S(x,y) \) are related by the partial differential equation

\[
\nabla \cdot \mathbf{F} = S \quad (C.1)
\]

Integrating (C.1) over an area \( RCD \), we obtain

\[
\int_{R} \left[ (F_1)_x + (F_2)_y \right] dx dy = \int_{R} S dx dy
\]

Application of Green's Theorem to the integral on the left yields the conservation relation

\[
\oint_{C} (F_1 dy - F_2 dx) = \int_{R} S dx dy \quad (C.2)
\]

where \( C \) is the boundary curve of Region \( R \) taken in the proper direction. In a physical sense the line integral represents a flux through the boundary of \( R \), and the integral over \( R \) a source within \( R \).

Now, in the transformed plane (C.1) becomes
\[
\frac{1}{J} \left\{ [y_\eta(F_1)_\xi - y_\eta(F_1)_\eta] + [x_\xi(F_2)_\eta - x_\eta(F_2)_\xi] \right\} = S \quad (C.3)
\]

But also note

\[
(y_\eta F_1 - x_\eta F_2)_\xi + (x_\xi F_2 - y_\xi F_1)_\eta = [y_\eta(F_1)_\xi - y_\eta(F_1)_\eta]
\]

\[+ [x_\xi(F_2)_\eta - x_\eta(F_2)_\xi] + [y_\xi F_1 - x_\xi F_2]
\]

\[+ [x_\xi F_2 - y_\xi F_1]
\]

which, since the last two terms cancel, implies that (C.3) becomes:

\[
(y_\eta F_1 - x_\eta F_2)_\xi + (x_\xi F_2 - y_\xi F_1)_\eta = S J
\]

Integrating over the area \(R^*\) in the transformed plane and applying Green's Theorem as before yields the relation

\[
\int_{\mathcal{C}^*} \left[ (y_\eta F_1 - x_\eta F_2) d\eta - (x_\xi F_2 - y_\xi F_1) d\xi \right] = \int_{R^*} S J d\xi d\eta \quad (C.4)
\]

where \(\mathcal{C}^*\) is the boundary curve of \(R^*\) (\(R^*\) and \(\mathcal{C}^*\) are the images of \(R\) and \(\mathcal{C}\) respectively). To see that (C.4) expresses the conservation relation in the transformed plane parallel to that expressed by (C.2) in the physical plane, consider calculating the components of \(F\) normal to lines of constant \(\xi\) and \(\eta\). Utilizing equations (A.26) and (A.28) we have

\[
F_\xi \cdot \eta^{(\eta)} = (x_\xi F_2 - y_\xi F_1)/\sqrt{\gamma}
\]

\[
F_\xi \cdot \eta^{(\xi)} = (y_\eta F_1 - x_\eta F_2)/\sqrt{\alpha}
\]
Let \( r \) be the position vector describing the points along the curve \( C^* \). Then, utilizing conventional techniques,

\[
|dr| = \sqrt{\gamma (d\xi)^2 + \beta d\xi d\eta + \alpha (d\eta)^2}
\]

Along a line of constant \( \eta \) we have

\[
|dr|_\eta = \sqrt{\gamma} \, d\xi
\]

and along a line of constant \( \xi \)

\[
|dr|_\xi = \sqrt{\alpha} \, d\eta
\]

Thus, the flux across a line of constant \( \eta \) is

\[
\mathbf{F} \cdot \mathbf{n}^{(\eta)} |dr|_\eta = (x_\xi F_2 - y_\xi F_1) d\xi
\]

and across a line of constant \( \xi \) becomes

\[
\mathbf{F} \cdot \mathbf{n}^{(\xi)} |dr|_\xi = (y_\eta F_1 - x_\eta F_2) d\eta
\]

These are the relations appearing in the flux terms of (C.4). The flux terms thus have an exact analog to those appearing in (C.2). Hence, the conservative relation (C.4) in the transformed plane expresses conservation in the physical plane over the non-square area formed by intersection of the curvilinear \( \xi \) and \( \eta \) coordinate lines in a manner which is precisely equivalent to the conservation expressed by (C.2) over the square area formed by the intersection of the \( x \) and \( y \) coordinate lines.
APPENDIX D

FINITE DIFFERENCE APPROXIMATIONS IN THE TRANSFORMED PLANE

This appendix contains a compilation of the second order finite difference expressions used to approximate partial derivatives in the transformed plane. Computational molecules for the derivative approximations appear to the right of each expression given. Combined difference forms for the transformation parameters $\alpha, \beta, \gamma, J, \sigma$, and $\tau$ are also included here. Since the field step size is immaterial in the $(\xi, \eta)$ plane, it is taken as unity for all approximations and does not appear explicitly. The following definitions are used throughout this appendix:

$f = f(\xi, \eta)$ - a twice continuously differentiable function of $\xi$ and $\eta$.

$x = x(\xi, \eta)$ - coordinate transformation function defined by Equation (13).

$y = y(\xi, \eta)$ - coordinate transformation function defined by Equation (13).

$\alpha, \beta, \gamma, J, \sigma$, and $\tau$ - transformation parameters defined in Appendix A.

In addition the following notational convention is utilized to indicate the position at which functions and derivatives are evaluated in the discrete $(\xi, \eta)$ plane:

$$f_{i,j} \equiv f(\xi_i, \eta_j), (f_x)_{i,j} \equiv f_{\xi}(\xi_i, \eta_j), (f_{\eta})_{i,j} \equiv f_{\eta}(\xi_i, \eta_j)$$
Derivative Approximations

First derivative, central differences:

\[(f^\xi)_i,j = \frac{(f^\xi)_{i+1,j} + (f^\xi)_{i-1,j}}{2}\]  
(D.1a)

where

\[(f^\xi)_{i+1,j} = f_{i+1,j} - f_{i-1,j}\]  
(D.1b)

\[(f^\eta)_{i+1,j} = \frac{(f^\eta)_{i+1,j+1} - (f^\eta)_{i+1,j-1}}{2}\]  
(D.2a)

where

\[(f^\eta)_{i+1,j+1} = f_{i+1,j+1} - f_{i+1,j-1}\]  
(D.2b)

First derivative, forward differences:

\[(f^\xi)_i,j = \frac{-f_{i+2,j} + 4f_{i+1,j} - 3f_{i,j}}{2}\]  
(D.3)

\[(f^\eta)_i,j = \frac{-f_{i+1,j+1} + 4f_{i+1,j} - 3f_{i,j}}{2}\]  
(D.4)

Second derivative, central differences:

\[(f^\xi\xi)_i,j = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{2} \approx (f^\xi)^i\]  
(D.5)

\[(f^\eta\eta)_i,j = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{2} \approx (f^\eta)^i\]  
(D.6)

\[(f^\xi\eta)_i,j = \frac{f_{i,j+1} - f_{i,j-1}}{2} \approx (f^\eta)^i\]  
(D.7a)

where
\[ f_{i+1,j+1} = f_{i+1,j-1} + f_{i-1,j-1} - f_{i-1,j+1} \] (D.7b)

**Transformation Parameters**

\[ a_{i,j} = a_{i,j}^{'}/4 \] (D.8a)

where

\[ a_{i,j}^{'},j \equiv (x_{\eta,i,j}^{'})^2 + (y_{\eta,i,j}^{'})^2 \] (D.8b)

\[ \beta_{i,j} = \beta_{i,j}^{'}/4 \] (D.9a)

where

\[ \beta_{i,j}^{'},j \equiv (x_{\xi,i,j}^{'})^2 + (y_{\xi,i,j}^{'})^2 \] (D.9b)

\[ \gamma_{i,j} = \gamma_{i,j}^{'}/4 \] (D.10a)

where

\[ \gamma_{i,j}^{'},j \equiv (x_{\xi,i,j}^{'})^2 + (y_{\xi,i,j}^{'})^2 \] (D.10b)

\[ J_{i,j} = J_{i,j}^{'}/4 \] (D.11a)

where

\[ J_{i,j}^{'},j \equiv (x_{\xi,i,j}^{'})^2 + (y_{\xi,i,j}^{'})^2 \] (D.11b)

\[ \sigma_{i,j} = \sigma_{i,j}^{'}/2 \] (D.12a)

where

\[ \sigma_{i,j}^{'},j \equiv [(y_{\xi,i,j}^{'})^2 (Dx_{\xi,i,j}^{'})^2 - (x_{\xi,i,j}^{'})^2 (Dy_{\xi,i,j}^{'})^2]/J_{i,j}^{'},j \] (D.12b)
\[
\tau_{i,j} = \frac{\tau_{i,j}'}{2}
\]  \hspace{1cm} (D.13a)

where

\[
\tau_{i,j}' = \left[ \left( x' \right)_{i,j} \left( D_{x'} \right)_{i,j} - \left( y' \right)_{i,j} \right] / \sqrt{J_{i,j}}
\]  \hspace{1cm} (D.13b)

and where

\[
\left( D_{x'} \right)_{i,j} = \alpha'_{i,j} \left( x' \xi \right)_{i,j} - 2\beta'_{i,j} \left( x' \zeta \right)_{i,j} + \gamma'_{i,j} \left( x' \eta \right)_{i,j}
\]  \hspace{1cm} (D.14a)

\[
\left( D_{y'} \right)_{i,j} = \alpha'_{i,j} \left( y' \xi \right)_{i,j} - 2\beta'_{i,j} \left( y' \zeta \right)_{i,j} + \gamma'_{i,j} \left( y' \eta \right)_{i,j}
\]  \hspace{1cm} (D.14b)
This appendix presents a detailed discussion of the methods used to determine contour plots of a function defined in the \( \xi, \eta \) plane. The numerical procedures which are used to transform the contour to a physical plane representation are also covered.

**Determination of Contours in the \( \xi, \eta \) Plane**

Let \( \phi = \phi(\xi, \eta) \) be a function defined in the region \( \overline{D^*} \) (Figure 1) possessing continuous second derivatives. Since \( \overline{D^*} \) is closed and bounded, let \( m(\phi) = \min \{ \phi(\xi, \eta) \mid [\xi, \eta] \in \overline{D^*} \} \) and \( M(\phi) = \max \{ \phi(\xi, \eta) \mid [\xi, \eta] \in \overline{D^*} \} \). If \( \phi \) is a number such that \( m(\phi) \leq \phi \leq M(\phi) \), then we define the \( \phi \)-contour of \( \phi(\xi, \eta) \), \( C_T(\phi) \), as the set

\[
C_T(\phi) = \{ [\xi, \eta] \mid [\xi, \eta] \in \overline{D^*} \text{ and } \phi(\xi, \eta) = \phi \}
\]

Graphically, \( C_T(\phi) \) is the curve created by the intersection of the graph of \( \phi(\xi, \eta) \) and the plane \( \phi(\xi, \eta) = \phi \). For plotting convenience the curve is usually projected onto the \( (\xi, \eta) \) plane. These ideas are illustrated in Figure E.1. The contour, \( C_T(\phi) \), is also often referred to as the level set of \( \phi(\xi, \eta) \) through \( \phi \).

Now suppose that \( \phi(\xi, \eta) \) is known only in a discrete fashion. That is, let the net function \( \phi_{i,j} = \phi(\xi_i, \eta_j) \) be known on the discrete set \( \overline{D^{**}} = \{ [\xi_i, \eta_j] \mid \xi_i = i-1 \text{ for } 1 \leq i \leq I_{\text{MAX}} \text{ and } \eta_j = j-1 \text{ for } 1 \leq j \leq J_{\text{MAX}} \} \)
Figure E.1. Contour - \( \phi \) of \( \phi(\xi, \eta) \)

(The fact that \( \phi_{i,j} \) may only be an approximation to \( \phi(\xi, \eta) \) is immaterial to the current discussion). If similar definitions for \( m(\phi) \) and \( M(\phi) \) are made for \( \phi_{i,j} \) on the set \( \overline{D}^{**} \), then the \( \phi \)-contour of \( \phi_{i,j} \), denoted \( C_T(\phi) \) again, can be defined as

\[
C_T(\phi) = \{ [\overline{\xi}_k, \overline{\eta}_k] \mid 0 \leq \overline{\xi}_k \leq \text{IMAX}-1; 0 \leq \overline{\xi}_k \leq \text{JMAX}-1; \\
\overline{\phi}(\overline{\xi}_k, \overline{\eta}_k) = \phi; k=1,2,\ldots,N; N \geq 2 \}
\]

The bars over \( \xi_k, \eta_k \), and \( \phi \) are to indicate that \( [\overline{\xi}_k, \overline{\eta}_k] \) may not be an element of \( \overline{D}^{**} \) and that \( \overline{\phi} \) is not necessarily one of the values of the net function \( \phi_{i,j} \). This is readily apparent from the discrete analog to Figure E.1 given in Figure E.2.
Figure E.2. Contour of $\phi(\xi, \eta_j)$

Numerically, the import of the above discussion is that interpolation between the points of the discrete set $D^{**}$ is required to determine $C_T(\phi)$. Consider a portion of grid $D^{**}$ as shown in Figure E.3a where $1 \leq I_1 < I_2 \leq IMAX$ and $1 \leq J_1 < J_2 \leq JMAX$. Each grid block is labeled by the $(i, j)$ coordinates of the lower left hand corner of the block. Block $(m, n)$ is shown on a larger scale in Figure E.3b.

In order to improve the plotted resolution consider subdividing each block into four triangles, as shown. The value of $\phi$ at $(m + \frac{1}{2}, n + \frac{1}{2})$ is taken as the four point average

$$\phi_{m + \frac{1}{2}, n + \frac{1}{2}} = (\phi_{m, n} + \phi_{m+1, n} + \phi_{m, n+1} + \phi_{m+1, n+1})/4$$

A local $(\xi, \eta)$ coordinate is affixed to each grid block as demonstrated
in Figure E.3b. In order to standardize the interpolation procedures, a local \((\xi, \mu)\) coordinate system is also placed on each of the sub-triangles as illustrated in the series of drawings given in Figure E.4.

Interpolation is carried out on each of the four triangles for each grid block in the segment \((I_1 \leq i \leq I_2, J_1 \leq j \leq J_2)\) of the set \(D^{**}\) specified. In particular interpolation is performed along each of the three sides of each of the triangles if the contour value, \(\phi\), lies between the values at the ends of the sides. Let \(d'\) be the directed distance from a given
triangle vertex to the point on the triangle side where the contour intersects that side (denoted by $\otimes$). This distance is illustrated in Figure E.3b for triangle 1 and is defined in an analogous manner for the other triangles. If $\phi_1$ is the value of $\phi_{i,j}$ at a particular triangle vertex and $\phi_2$ the value of $\phi_{i,j}$ at the other end of the side, then $d'$ may be expressed as

\[ d' = (\text{side length})(\phi - \phi_1)/(\phi_2 - \phi_1) \]

For example, along side 1-2 of triangle 1, $d'$ is given by

\[ d'_{1-2} = (1.0)(\phi - \phi_{m+1,n})/(\phi_{m,n} - \phi_{m+1,n}) \]

Noting that the sides of the triangle have lengths 1.0, 1.0/$\sqrt{2}$, and 1.0/$\sqrt{2}$, the contour intersections can be expressed in the local triangle coordinates as

<table>
<thead>
<tr>
<th>Side</th>
<th>$\zeta_\ell$</th>
<th>$\mu_\ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>1-d</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>$d/2$</td>
<td>$d/2$</td>
</tr>
<tr>
<td>3-1</td>
<td>$(1+d)/2$</td>
<td>$(1-d)/2$</td>
</tr>
</tbody>
</table>

where $d \equiv (\phi - \phi_1)/(\phi_2 - \phi_1)$ and where $\ell=1,2,3,$ or 4 denotes the triangle number.

Once the contour intersections have been determined in the local triangle coordinates, $(\zeta_\ell, \mu_\ell)$, they must be transformed to the grid block coordinates $(\xi_m, \eta_m, n_m, n)$. This is done in the conventional fashion using orthogonal rotation matrices. If $[\zeta_\ell, \mu_\ell, \phi_{i,j}]$ are the coordinates of an intersection in triangle $\ell$, then
\[
\begin{bmatrix}
(x_m, n)_{\xi, p} \\
(y_m, n)_{\xi, p}
\end{bmatrix} = \begin{bmatrix} A_\xi \\
\mu_{\xi, p}
\end{bmatrix}
\]

where

\[
A_1 = \begin{bmatrix} 1 & 0 \\
0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\
-1 & 0 \end{bmatrix}
\]

\[
A_3 = \begin{bmatrix} -1 & 0 \\
0 & -1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & -1 \\
1 & 0 \end{bmatrix}
\]

Note that up to three contour intersections can occur for each triangle (i.e., one on each side). Finally, the point \([(x_m, n)_{\xi, p}, (y_m, n)_{\xi, p}] \) is transformed to \((\xi, \eta)\) coordinates by a simple linear transformation producing an element \([\xi_k, \eta_k]\) of the set \(C_T(\phi)\).

Transformation to the Physical Plane

Since contours in the \((\xi, \eta)\) plane are of little interest, \(C_T(\phi)\) must be transformed to the physical plane. This is made possible through the use of the coordinate transformation functions \(x(\xi_i, \eta_j)\) and \(y(\xi_i, \eta_j)\). Again interpolation is required since almost all elements of \(C_T(\phi)\) are not elements of \(D^{**}\) on which the discrete functions \(x(\xi_i, \eta_j)\) and \(y(\xi_i, \eta_j)\) are defined. As illustrated in Figure E.5 this implies a double linear interpolation must be performed. If \([\xi_k, \eta_k]\) denotes an element of \(C_T(\phi)\), the first step is to locate the \(\xi\) and \(\eta\) values bracketing \(\xi_k\) and \(\eta_k\). Denoting these by \(\xi_i, \xi_{i+1}\) and \(\eta_j, \eta_{j+1}\) as shown in the figure, the values of \(\bar{x_k}\) and \(\bar{y_k}\) are calculated as follows

\[
\bar{x_k} = (\bar{\eta_k} - \eta_j)(X_{j+1} - X_j)/(\eta_{j+1} - \eta_j) + X_j
\]

\[
\bar{y_k} = (\bar{\xi_k} - \xi_i)(Y_{i+1} - Y_i)/(\xi_{i+1} - \xi_i) + Y_i
\]
where

\[ X_j = (\xi_k - \xi_1)(x_{i+1,j} - x_{i,j})/(\xi_{i+1} - \xi_1) + x_{i,j} \]

\[ X_{j+1} = (\xi_k - \xi_1)(x_{i+1,j+1} - x_{i,j+1})/(\xi_{i+1} - \xi_1) + x_{i,j+1} \]

for all \( k = 1, 2, \ldots, N \). Similar expressions are used to calculate \( \bar{y}_k \).

Figure E.5. Interpolation for \( x \) and \( y \)