DOCUMENTATION OF PROGRAM COORDC TO GENERATE COORDINATE SYSTEM FOR 3-D CORNERS WITH OR WITHOUT FILLET USING BODY-FITTED CURVILINEAR COORDINATES—Part II

Dilip Kumar

SYSTEMS AND APPLIED SCIENCES CORPORATION
Hampton, Virginia 23666

Because of its significant early commercial potential, this information which has been developed under a U. S. Government contract is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations. Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.

DATE OF GENERAL RELEASE JUNE 1982.

NASA Contract NASl-15604
June 1980

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>List of Symbols</td>
<td>ii</td>
</tr>
<tr>
<td>I. Numerical Generation of Body-Fitted Curvilinear Coordinate Systems</td>
<td>1</td>
</tr>
<tr>
<td>A. Mathematics of Transformation</td>
<td>2</td>
</tr>
<tr>
<td>B. Numerical Solution Technique</td>
<td>6</td>
</tr>
<tr>
<td>C. Initial Guess and Boundary Point Specification</td>
<td>9</td>
</tr>
<tr>
<td>D. X-Coordinate and Metric Coefficients</td>
<td>9</td>
</tr>
<tr>
<td>E. Concluding Remarks</td>
<td>11</td>
</tr>
<tr>
<td>II. Computer Program</td>
<td>14</td>
</tr>
<tr>
<td>A. Listing of the Computer Program</td>
<td>15</td>
</tr>
<tr>
<td>B. Job Control Cards</td>
<td>29</td>
</tr>
<tr>
<td>C. Input Explanation</td>
<td>29</td>
</tr>
<tr>
<td>D. Sample Input</td>
<td>30</td>
</tr>
<tr>
<td>E. Sample Output</td>
<td>34</td>
</tr>
<tr>
<td>III. Programmer/Analyst Section</td>
<td>49</td>
</tr>
<tr>
<td>A. Main Program COORDC</td>
<td>49</td>
</tr>
<tr>
<td>B. Subroutine CALCOR</td>
<td>50</td>
</tr>
<tr>
<td>C. Subroutine IPRTC</td>
<td>51</td>
</tr>
<tr>
<td>D. Subroutine IPLTC</td>
<td>51</td>
</tr>
<tr>
<td>E. Block Diagram</td>
<td>51</td>
</tr>
</tbody>
</table>

N80-74862
ABSTRACT

The computer program COORDC generates a body-fitted curvi-linear coordinate system for corner geometry with or without corner fillets. It is assumed that at any given $\xi$, $x$ remains constant; consequently the only variation is in $y$ and $z$. It is also assumed that for all $\xi$'s in the physical plane the coordinate system in $y$-$z$ plane is similar. This enables solution of coordinate system for one particular $\xi = 1$ ($x$ for $\xi = 1$ is arbitrarily chosen to be 0.0) and the solution for all other $\xi$ plane can be easily specified once the coordinates in the physical plane on the line $1 \leq \xi \leq \text{IMAX}$, $\eta = 1$, $\zeta = 1$ are specified.
List of Symbols

**IMAX** Maximum number of points in $\xi$ - direction

**JMAX** Maximum number of points in $\eta$ - direction

**KMAX** Maximum number of points in $\zeta$ - direction

**J** Jacobian

**P, Q, R** Inhomogeneous terms used for coordinate attraction

**S, O, R** Successive-Over-Relaxation iteration

**X, Y, Z** Physical Coordinates

**\(a, \beta, \gamma\)** Metric Coefficients

**\(\xi, \eta, \zeta\)** Transformed Coordinates

**\(\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\)** Boundary contours in the physical plane

**\(\Gamma_1^*, \Gamma_2^*, \Gamma_3^*, \Gamma_4^*\)** Boundary contours in the transformed plane

**\(\omega\)** Acceleration parameter for Gauss-Seidel iteration

**Subscripts**

\(\xi, \eta, \zeta, x, y, z\) Denotes first partial differentiation

\(\xi\xi, \eta\eta, xx, yy\) etc. Denotes second partial differentiation

\(\xi\eta, xy, xz\) etc. Denotes cross partial differentiation

**Superscripts**

\((s)\) Denotes the iteration number
I. Numerical Generation of Body-Fitted Curvilinear Coordinate Systems

The method for numerically generating boundary-fitted curvilinear coordinate systems is presented as applicable in a specialized 3-D corner with or without a fillet. The coordinate system is generated such that for a specified \( \xi \), \( x \) remains constant and for all \( \xi \)'s the solution in \( y-z \) plane stays similar. This is particularly beneficial because for \( \xi = 1 \) plane (\( x \) is chosen to be 0.0 for \( \xi = 1 \)) \( y-z \) solution can be obtained numerically by the 2-D formulation instead of 3-D formulation of the body-fitted curvilinear coordinate system. The coordinates for all other \( \xi \) planes can be easily obtained once the physical plane coordinates on the line \( \xi = 1, \) IMAX, \( \eta = 1, \) \( \zeta = 1 \) are specified. In the present problem it is assumed that for a given \( \xi \) plane, right boundary and upper boundary have \( z \) and \( y \) constant respectively. In addition, on the left wall and the bottom wall, \( z \) and \( y \) respectively equal zero some distance from the corner when fillet is present; in case of no fillet \( z \) and \( y \) on left and bottom wall respectively is equal to zero. The clustering of \( \eta \) and \( \zeta \) lines is specified by the point distribution on \( \eta = 1 \) and \( \zeta = 1 \) line.

Section A provides a brief discussion of the mathematical formulation and section B describes the numerical technique used to generate the coordinate system. Section C explains how the required initial guess is specified. Section D deals with handling of the \( x \)-coordinate and the metric coefficients.
A. Mathematics of Transformation

Consider the transformation of a simply connected region $\mathbb{R}$, into a rectangular region as shown in Figure 1. We require that $\Gamma_1$, $\Gamma_2$, $\Gamma_3$, $\Gamma_4$ map into $\Gamma_1^*$, $\Gamma_2^*$, $\Gamma_3^*$, $\Gamma_4^*$ respectively. For identification purposes, region $\mathbb{D}$ is referred to as the physical plane, and $\mathbb{D}^*$ as the transformed plane. In the physical plane it is assumed that $x$ is constant for a specified $\xi$, $z$ is constant on $\Gamma_3$, and $y$ is constant on $\Gamma_4$. If for $\xi = 1$, $x = 0.0$ is assumed, then the transformation from $x, y, z$ to $\xi, \eta, \zeta$ is defined as follows,

\[
\begin{bmatrix}
\xi \\
\eta \\
\zeta
\end{bmatrix} =
\begin{bmatrix}
\xi(x) \\
\eta(y, z) \\
\zeta(y, z)
\end{bmatrix}
\]  

(1)

The above equation is solved only for $\eta$ and $\zeta$ since $\xi$ is explicitly known. Assuming that an inverse transformation exists, then

\[
\begin{bmatrix}
y \\
z
\end{bmatrix} =
\begin{bmatrix}
y(\eta, \zeta) \\
z(\eta, \zeta)
\end{bmatrix}
\]  

(2)

where the jacobian $J$ is as follows:

\[
J = y_\eta z_\zeta - y_\zeta z_\eta
\]  

(3)

The generating elliptic system is chosen to be the inhomogeneous Laplace equation,

\[
\eta_{yy} + \eta_{zz} = Q(\eta, \zeta)
\]  

(4a)

\[
\zeta_{yy} + \zeta_{zz} = R(\eta, \zeta)
\]  

(4b)
with the boundary conditions as follows:

on $\Gamma_1$

\[
\begin{bmatrix}
\eta \\
\zeta
\end{bmatrix} = \begin{bmatrix}
\eta_1(y,z) \\
\zeta_1
\end{bmatrix}
\]  

(5a)

on $\Gamma_2$

\[
\begin{bmatrix}
\eta \\
\zeta
\end{bmatrix} = \begin{bmatrix}
\eta_2(y,z) \\
\zeta_2(y,z)
\end{bmatrix}
\]  

(5b)

on $\Gamma_3$

\[
\begin{bmatrix}
\eta \\
\zeta
\end{bmatrix} = \begin{bmatrix}
\eta_3(y,z) \\
\zeta_3
\end{bmatrix}
\]  

(5c)

and on $\Gamma_4$

\[
\begin{bmatrix}
\eta \\
\zeta
\end{bmatrix} = \begin{bmatrix}
\eta_4(y,z) \\
\zeta_4(y,z)
\end{bmatrix}
\]  

(5d)

where $\zeta_1$, $\eta_2$, $\zeta_3$, $\eta_4$ are specified constants and $\eta_1$, $\zeta_2$, $\eta_3$, $\zeta_4$ are specified functions.

The inhomogeneous terms $Q$ and $R$ are selected to control the spacing of $\eta = \text{constant}$ and $\zeta = \text{constant}$ lines in the physical plane; several forms for the inhomogeneous terms can be used. In the present case a special form of $Q$ and $R$ terms was used based on the point distribution on $\eta = \eta_2$ and $\zeta = \zeta_1$ line. (For convenience $\eta_2 = \zeta_1 = 1$ and $\eta_4 = J\text{MAX}$ and $\zeta_4 = K\text{MAX}$ is chosen thus ensuring a field size of $J\text{MAX} \times K\text{MAX}$ for a given $\xi$ plane).

In order that the transformed plane contain a uniform grid of spacing equal to unity, the dependent and independent variables
must be interchanged in equations (4a) and (4b). The resulting equations after transformation are as follows:

\[
\begin{align*}
\alpha Y_{n\eta} & - 2\beta Y_{n\zeta} + \gamma Y_{\zeta^2} = -J^2 \left[ Y_{n\eta} Q(n,\zeta) + Y_{\zeta} R(n,\zeta) \right] \quad \text{(6a)} \\
\alpha Z_{n\eta} & - 2\beta Z_{n\zeta} + \gamma Z_{\zeta^2} = -J^2 \left[ Z_{n\eta} Q(n,\zeta) + Z_{\zeta} R(n,\zeta) \right] \quad \text{(6b)}
\end{align*}
\]

where,
\[
\begin{align*}
\alpha & = y_{\zeta^2} + z_{\zeta^2} \\
\beta & = y_{n\eta} y_{n\zeta} + z_{n\eta} z_{n\zeta} \\
\gamma & = y_{\eta}^2 + z_{\eta}^2 \\
J & = y_{n\eta} z_{\zeta} - y_{\zeta} z_{n}
\end{align*}
\]

The boundary conditions transform as follows:

on \(\Gamma_1^*\)
\[
\begin{bmatrix} y \\ z \end{bmatrix} \begin{bmatrix} a_1(n,\zeta_1) \\ a_2(n,\zeta_1) \end{bmatrix}
\]

\quad \text{(7a)}

on \(\Gamma_2^*\)
\[
\begin{bmatrix} y \\ z \end{bmatrix} \begin{bmatrix} b_1(n_2,\zeta) \\ b_2(n_2,\zeta) \end{bmatrix}
\]

\quad \text{(7b)}

on \(\Gamma_3^*\)
\[
\begin{bmatrix} y \\ z \end{bmatrix} \begin{bmatrix} c_1(n,\zeta_3) \\ c_2(n,\zeta_3) \end{bmatrix}
\]

\quad \text{(7c)}

and on \(\Gamma_4^*\)
\[
\begin{bmatrix} y \\ z \end{bmatrix} \begin{bmatrix} d_1(n_4,\zeta) \\ d_2(n_4,\zeta) \end{bmatrix}
\]

\quad \text{(7d)}

The functions \(a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\) are specified.
by known contours $\Gamma_1$, $\Gamma_2$, $\Gamma_3$, $\Gamma_4$. If the coordinate point locations on these contours is fixed then Dirichlet type boundary conditions can be used. On the other hand if the coordinate lines on these contours is to be made parallel or normal to some line then Neumann type conditions can be used. On $\Gamma_1^*$ and $\Gamma_2^*$ Dirichlet type conditions were used. On $\Gamma_3^*$ and $\Gamma_4^*$ Neumann type conditions were used. Thus the following expressions are obtained on $\Gamma_3^*$ and $\Gamma_4^*$:

On $\Gamma_3^*$

\begin{align}
  y(J,K_{\text{MAX}}) &= y(J,K_{\text{MAX}}-1) \\
  z(J,K_{\text{MAX}}) &= \text{constant } z_1 \\
  \text{for } 1 < J < J_{\text{MAX}}
\end{align}

and on $\Gamma_4^*$

\begin{align}
  z(J_{\text{MAX}},K) &= z(J_{\text{MAX}}-1,K) \\
  y(J_{\text{MAX}},K) &= \text{constant } y_1 \\
  \text{for } 1 < K < K_{\text{MAX}}
\end{align}

The last aspect to be discussed is how $Q$ and $R$ are computed. $Q$ is computed based on the $y$ variation of points on $\Gamma_1$ and $R$ is computed based on the $z$ variation of points on $\Gamma_2$. The following discussion will make this point clearer.

If one assumes on $\Gamma_1$ that coordinate points are specified such that the $y$ derivatives with respect to $\zeta$ are zero then from (6a),

\[ Q(\eta,\zeta) = -\frac{Y_{\eta\eta}}{Y_{\eta}} \]

Also, $Q$ is not a function of $\zeta$. Thus, only one one-dimensional array of size $J_{\text{MAX}}$ is sufficient to store $Q$. Similarly for $R$ we obtain,

\[ z_{\zeta\zeta} \]
Again R is not a function of $\eta$. A one-dimensional array of size $K_{\text{MAX}}$ is sufficient to store $R$.

B. Numerical Solution Technique

Based on the above discussion, the following equations must be numerically solved:

\begin{align*}
\alpha y_{\eta\eta} - 2 \beta y_{\eta\zeta} + \gamma y_{\zeta\zeta} &= - J^2 [y_{\eta} Q(\eta) + y_{\zeta} R(\zeta)] \quad (10a) \\
\alpha z_{\eta\eta} - 2 \beta z_{\eta\zeta} + \gamma z_{\zeta\zeta} &= - J^2 [z_{\eta} Q(\eta) + z_{\zeta} R(\zeta)] \quad (10b)
\end{align*}

where

\begin{align*}
Q(\eta) &= - \frac{y_{\eta\eta}}{y_\eta} \quad \text{on } \Gamma_1 \quad (11a) \\
R(\zeta) &= - \frac{z_{\zeta\zeta}}{z_\zeta} \quad \text{on } \Gamma_2 \quad (11b)
\end{align*}

The first and second order derivatives in (10) and (11) are written in central difference form. The expressions for $Q$ and $R$ are as follows:

\begin{align*}
Q(J) &= - \frac{y(J+1,1) - 2y(J,1) + y(J-1,1)}{(0.5(y(J+1,1) - y(J-1,1)))^{**3}} \quad 2 < J < \text{JMAX-1} \quad (12a) \\
R(K) &= - \frac{z(1,K+1) - 2z(1,K) + z(1,K-1)}{(0.5(z(1,K+1) - z(1,K-1)))^{**3}} \quad 2 < K < \text{KMAX-1} \quad (12b)
\end{align*}

Equations (10a) and (10b) are solved by successive over
relaxation (S.O.R) technique. The intermediate value for \(y\) and \(z\) are as follows:

\[
\bar{Y}(J,K) = \alpha (y(J+1,K) + y(J-1,K)) - 2\beta y_{\eta\zeta} + \gamma (y(J,K+1) + y(J,K-1)) + J^2 (y_{\eta} Q(J) + y_{\zeta} R(K)) / [2(\alpha + \gamma)]
\] (13a)

where

\[
y_{\eta\zeta} = 0.25 \left[ y(J+1,K+1) - y(J+1,K-1) - y(J-1,K+1) + y(J-1,K-1) \right]
\] (13b)

\[
y_{\eta} = 0.5 \left[ y(J+1,K) - y(J-1,K) \right]
\] (13c)

\[
y_{\zeta} = 0.5 \left[ y(J,K+1) - y(J,K-1) \right]
\] (13d)

\[
\alpha = 0.25 \left[ (y(J+1,K+1) - y(J,K-1))^2 + (z(J+1,K+1) - z(J,K-1))^2 \right]
\] (13e)

\[
\beta = 0.25 \left[ (y(J+1,K) - y(J-1,K)) (y(J,K+1) - y(J,K-1)) + (z(J+1,K) - z(J-1,K)) (z(J,K+1) - z(J,K-1)) \right]
\] (13f)

\[
\gamma = 0.25 \left[ (y(J+1,K) - y(J-1,K))^2 + (z(J+1,K) - z(J-1,K))^2 \right]
\] (13g)

\[
J = 0.25 \left[ (y(J+1,K) - y(J-1,K)) (z(J+1,K) - z(J,K-1)) - (y(J,K+1) - y(J,K-1)) (z(J+1,K) - z(J-1,K)) \right]
\] (13h)

similarly an expression for \(\bar{z}\) can be written out. For each grid
point two finite difference equations (one for \( y \) and one for \( z \)) are obtained. The subscripts \( J \) and \( K \) have a range as follows:

\[
J = 2, 3, 4, \ldots, J\text{MAX}-1
\]

\[
K = 2, 3, 4, \ldots, K\text{MAX}-1
\]

This results in a set of \((J\text{MAX}-2)(K\text{MAX}-2)\). The above equations are solved simultaneously using S.O.R. For a set of non-linear differential equations for some variable \( f \), S.O.R. iteration can be written as follows:

\[
f^{(s+1)} = w \overline{f} + (1 - w) f^{(s)}
\]

where superscripts denote the iteration number, \( w \) is the acceleration parameter, and the latest values are used in the difference equations to solve \( \overline{f} \). In the present problem \( f \) denotes \( y \) or \( z \).

Since on \( \Gamma_3 \) and \( \Gamma_4 \), Neumann type boundary conditions are applied, these conditions are implemented as follows:

on \( \Gamma_3 \)

\[
y(J, K\text{MAX}) = y(J, K\text{MAX}-1)
\]

\[
2 \leq J \leq J\text{MAX}
\]

on \( \Gamma_4 \)

\[
z(J\text{MAX}, K) = z(J\text{MAX}-1, K)
\]

\[
2 \leq K \leq K\text{MAX}
\]

Since on \( \Gamma_3 \) \( z = \text{constant} \) and on \( \Gamma_4 \) \( y = \text{constant} \), equations \((15a)\) and \((15b)\) ensure that coordinate lines on \( \Gamma_3 \) and \( \Gamma_4 \) are orthogonal even though the coordinate system is not necessarily orthogonal everywhere.
C. Initial Guess and Boundary Point Specification

The input to the program consists of points specification on $\Gamma_1$ and $\Gamma_2$ boundary. $\Gamma_1$ boundary is such that the variation of $z$ close to the $J_{\text{MAX}}$ point is zero. Similarly on $\Gamma_2$ the $y$ variation close to $K_{\text{MAX}}$ point is zero. Points on $\Gamma_1$ and $\Gamma_2$ are specified implied that $\{y(J, l), z(J, l)\}$ $1 < J < J_{\text{MAX}}$ and $\{y(l, K), z(l, K)\}$ $1 < K < K_{\text{MAX}}$ are known. Then for the initial guess we have,

\[
z(J, K) = z(l, K_{\text{MAX}}), \quad 2 < J < J_{\text{MAX}} \tag{16a}
\]
\[
y(J_{\text{MAX}}, K) = y(J_{\text{MAX}}, l), \quad 2 < K < K_{\text{MAX}} \tag{16b}
\]

also,

\[
y(J, K) = y(l, 1) + (y(J_{\text{MAX}}, K) - y(l, 1)) \times \left(\frac{y(J, 1) - y(l, 1)}{y(J_{\text{MAX}}, 1) - y(l, 1)}\right) \tag{16c}
\]
\[
z(J, K) = z(J, 1) + (z(J, K_{\text{MAX}}) - z(J, 1)) \times \left(\frac{z(l, 1)}{z(l, K_{\text{MAX}}) - z(l, 1)}\right) \tag{16d}
\]

$2 < J < J_{\text{MAX}} - 1$ and $2 < K < K_{\text{MAX}} - 1$

and

\[
y(J, K_{\text{MAX}}) = y(J, K_{\text{MAX}} - 1), \quad \leq J < J_{\text{MAX}} - 1 \tag{16e}
\]
\[
z(J_{\text{MAX}}, K) = z(J_{\text{MAX}} - 1, K), \quad \leq K < K_{\text{MAX}} - 1 \tag{16f}
\]

D. x-Coordinate and Metric Coefficients

As discussed before, the $y$ and $z$ variation is computed only for one plane $\xi = 1$; also $x$ is constant for any specified $\xi$-plane. Thus if $x, y, z$ variation on $1 < \xi < \text{IMAX}$, $\eta = 1$ and $\zeta = 1$ is specified then coordinates of any point can be easily specified. For example say for $\xi = 1$,
\[ x(Il,l,l) = xl \]  
\[ y(Il,l,l) = yl \]  
\[ z(Il,l,l) = zl \]  

Then for any point \( Il, J, K \) in \( \xi = Il \) plane we have,

\[ x(Il,J,K) = xl \]  
\[ y(Il,J,K) = y(Il,l,l) - y(l,l,l) + y(l,J,K) \]  
\[ z(Il,J,K) = z(Il,l,l) - z(l,l,l) + z(l,J,K) \]

If the coordinate system on \( \xi = 1 \) plane is continuous in \( y \) and \( z \), and if on \( \xi = 1, \text{IMAX}, \eta = 1, \zeta = 1 \) line the variation in \( x, y, z \) is continuous then the solution given by above equations will be continuous. This is ensured by having continuous \( y \) and \( z \) variation on \( \Gamma_1 \) and \( \Gamma_2 \) and on line \( \xi = 1, \text{IMAX}, \eta = 1, \zeta = 1 \).

A brief discussion on metric coefficients can now be included. The nine metric coefficients \( \beta_{ij} \) are given by:

\[ \beta_{11} = \frac{(y_\eta z_\zeta - y_\zeta z_\eta)}{J} \]  
\[ \beta_{12} = \frac{(y_\zeta z_\eta - y_\eta z_\zeta)}{J} \]  
\[ \beta_{13} = \frac{(y_\xi z_\eta - y_\eta z_\xi)}{J} \]  
\[ \beta_{21} = \frac{(x_\zeta z_\eta - x_\eta z_\zeta)}{J} \]  
\[ \beta_{22} = \frac{(x_\zeta z_\zeta - x_\zeta z_\xi)}{J} \]  
\[ \beta_{23} = \frac{(x_\eta z_\zeta - x_\xi z_\eta)}{J} \]
\[ \beta_{31} = \frac{(x_\eta y_\zeta - x_\zeta y_\eta)}{J} \quad (18g) \]
\[ \beta_{32} = \frac{(x_\zeta y_\xi - x_\xi y_\zeta)}{J} \quad (18h) \]
\[ \beta_{33} = \frac{(x_\xi y_\eta - x_\eta y_\xi)}{J} \quad (18i) \]

Since \( x \) is constant for any specified \( \xi \) plane, we have \( x_\eta = x_\zeta = 0 \) which considerably simplifies the above equations. Also, we can ensure that \( y_\xi = 0, z_\xi = 0 \) if \( y \) and \( z \) are specified to be constant on \( 1 \leq \xi \leq \text{IMAX}, \eta = 1, \zeta = 1 \) line. In such a case the metric coefficients become,

\[ \beta_{11} = \frac{(y_\eta z_\zeta - y_\zeta z_\eta)}{J} \quad (19a) \]
\[ \beta_{12} = 0.0 \quad (19b) \]
\[ \beta_{13} = 0.0 \quad (19c) \]
\[ \beta_{21} = 0.0 \quad (19d) \]
\[ \beta_{22} = \frac{x_\xi z_\zeta}{J} \quad (19e) \]
\[ \beta_{23} = -\frac{x_\xi z_\eta}{J} \quad (19f) \]
\[ \beta_{31} = 0.0 \quad (19g) \]
\[ \beta_{32} = -\frac{x_\xi y_\zeta}{J} \quad (19h) \]
\[ \beta_{33} = \frac{x_\xi y_\eta}{J} \quad (19i) \]
\[ J = \frac{x_\xi (y_\eta z_\zeta - y_\zeta z_\eta)}{J} \quad (19j) \]

E. Concluding Remarks

A method for generating 3-D body-fitted coordinate systems for corner flows has been presented. The computer code to generate coordinate systems using equations 12, 13, 14, 15, 16 is presented in Chapter 3. The generated coordinate system is stored on a disk file which becomes a part of the input to the Navier Stokes code. Successful coordinate systems for corners
with or without fillet and for a 3-D flat plate have been generated (the program can generate suitable coordinate systems for a 3-D flat plate problem). The compressible laminar Navier Stokes solution for the above mentioned problems at low Reynolds number have already been obtained. Effort is now underway to obtain solution for higher Reynolds number with or without turbulence.
\[ z = \text{constant close to } J_{\text{MAX}} \text{ on } \Gamma_1 \]

\[ y = \text{constant on } \Gamma_4 \]

\[ z = \text{constant on } \Gamma_3 \]

\[ y = \text{constant close to } K_{\text{MAX}} \text{ on } \Gamma_2 \]

**Physical Plane**

**Transformed Plane**
II. Computer Program

A. Listing of the Computer Program

The program listing follows this page.
CREATION RUN

CARDS ENCOUNTERED IN INPUT

** +DECK CORD1
** PROGRAM CORDC(INPUT, OUTPUT, SOLNXX, TAPE1=SOLNXX, TAPE5=INPUT,
** TAPE6=OUTPUT)

C**
C* CORNER COORDINATE GENERATION PROGRAM BASED ON METHOD
C* DEVELOPED BY THOMPSON, THAMES, AND MASTIN OF MISSISSIPPI
C* STATE UNIVERSITY.
C*
C** FOR FURTHER ENQUIRIES CONTACT :
C*
C** DR. JULIUS HARRIS
C** DR. DILIP KUMAR
C** NASA - LANGLEY RESEARCH CENTER
C** MAIL STOP 163
C** HAMPTON, VA. 23665
C** PHONE: 804-827-3696
C** FTS CODE 928-1110
C*

C**
C* THIS PROGRAM GENERATES COORDINATE SYSTEM FOR A 2-D CORNER
C* WITH OR WITHOUT FILLER. THE METHOD USED IS BASED ON METHOD
C* DEVELOPED BY THOMPSON, THAMES, MASTIN AND OTHERS AT MISS.
C* STATE UNIVERSITY. THE CORNER IS SUCH THAT EXCEPT IN THE
C* FILLER (WHEN PRESENT) L.H.S. BOUNDARY HAS Z = 0 AND BOTTOM
C* SIDE BOUNDARY HAS Y = 0. ALSO R.H.S. BOUNDARY ALWAYS HAS
C* Z = CONSTANT AND UPPER BOUNDARY HAS Y = CONSTANT. THE
C* PROGRAM GENERATES Q AND R SUCH THAT COORDINATE CLUSTERING
C* IS PROPORTIONAL TO COORDINATE POINTS DISTRIBUTION ON L.H.S.
C* AND BOTTOM BOUNDARY. ON R.H.S. AND TOP BOUNDARIES NEUMANN
C* TYPE BOUNDARY CONDITIONS ARE USED. THIS ENSURES THAT
C* COORDINATE LINES ON R.H.S. AND UPPER BOUNDARY ARE PARALLEL
C* TO SIDES.

--- CARD INPUT ---

C**
C** CORD1 SPECIFIES FIELD SIZE AND PRINT, PLOT Flags
C** JMAX, KMAX, IPRT1, IPRT2, IFLT1, IFLT2, ITERMX, R1, R2, R3
C** (FORMAT 715, 5X, 3F10.0)
CARDS ENCOUNTERED IN INPUT

JMAX  NO OF NODES IN Y-DIRECTION
KMAX  NO OF NODES IN Z-DIRECTION
IPRT1 INITIAL GUESS IS TO BE PRINTED OR NOT
   IF IPRT1 = 1 THEN PARTIALLY CONVERGED
      SOLUTION IS PRINTED AFTER 50 ITER.
IPRT2 FINAL SOLUTION TO BE PRINTED OR NOT
IPLT1 INITIAL GUESS TO BE PLOTTED ON NOT
IPLTZ FINAL SOLUTION TO BE PLOTTED ON NOT
ITERMX MAXIMUM NO. OF ITERATIONS ALLOWED
R1 GAUSS-SEIDEL ITERATION PARAMETER
(USUALLY 0.6)
R2 CONVERGENCE CRITERION IN Y-DIRECTION
   (USUALLY 0.000005 TO 0.00001)
R3 CONVERGENCE CRITERION IN Z-DIRECTION
   (USUALLY SAME AS RZ)

NOTE: FOR ALL FLAGS 0 MEANS NO, 1 MEANS YES

CARD1 CARDS TO SPECIFY POINT DISTRIBUTION ON L.H.S.
   BOUNDARY (Y-DIRECTION). NO. OF CARDS  =  JMAX
   T,Z  (FORMAT 2F10.0)
   Y   1 Y COORDINATE
   Z   1 Z COORDINATE

CARD3 CARDS TO SPECIFY POINT DISTRIBUTION ON BOTTOM
   BOUNDARY (Z-DIRECTION). NO. OF CARDS  =  KMAX
   T,Z  (FORMAT 2F10.0) SEE CARD TYPE 2

CARD4 SPECIFIES NO. OF NODES IN X-DIRECTION.
   IMAX  (FORMAT I5)
   IMAX  NO. OF POINTS IN X-DIRECTION

CARD5 CARDS TO SPECIFY NO. OF POINTS IN X - DIRECTION.
   X,Y1,Z1  (FORMAT 3F10.0)
   X  1 X-COORDINATE ON J=1,K=1 LINE (X-DIRECTION)
   Y1  1 Y-COORDINATE ON J=1,K=1 LINE (X-DIRECTION)
   Z1  1 Z-COORDINATE ON J=1,K=1 LINE (X-DIRECTION)

CARD6 PRINT/Plot HEADINGS.
   HED1(4)  (FORMAT 4A10)
   HED1  1 40 BYTES OF PLOT HEADING

CARD7 ADDITIONAL PRINT/Plot HEADINGS.
   HED2(4)  (FORMAT 4A10)
   HED2  1 40 BYTES OF PLOT HEADING

CAROZ CARDS TO SPECIFY POINT DISTRIBUTION ON L.H.S.
BOUNDARY (Y-DIRECTION). NO. OF CARDS  =  JMAX

CAROZ CARDS TO SPECIFY POINT DISTRIBUTION ON BOTTOM
BOUNDARY (Z-DIRECTION). NO. OF CARDS  =  KMAX

CARD8 CARDS TO SPECIFY NO. OF NODES IN X-DIRECTION.
   IMAX  (FORMAT I5)
   IMAX  NO. OF POINTS IN X-DIRECTION

CARD9 CARDS TO SPECIFY NO. OF POINTS IN X - DIRECTION.
   X,Y1,Z1  (FORMAT 3F10.0)
   X  1 X-COORDINATE ON J=1,K=1 LINE (X-DIRECTION)
   Y1  1 Y-COORDINATE ON J=1,K=1 LINE (X-DIRECTION)
   Z1  1 Z-COORDINATE ON J=1,K=1 LINE (X-DIRECTION)

CARD10 PRINT/Plot HEADINGS.
   HED1(4)  (FORMAT 4A10)
   HED1  1 40 BYTES OF PLOT HEADING

CARD11 ADDITIONAL PRINT/Plot HEADINGS.
   HED2(4)  (FORMAT 4A10)
   HED2  1 40 BYTES OF PLOT HEADING
CARDS ENCOUNTERED IN INPUT

HED2 : 40 BYTES OF ADDITIONAL PLOT HEADING.

KVIS2,T777,C70000.
USER,479019C.
CHARGE,1007188,1RC.
GETQDPL,KHRCO1,UN=375732N.
UPDATE(F,C=VISFL1).
COPY8R,INPUT,EDTFI1.
REWRITEVISFL1,EDTFI1.
EDIT,VISFL1,1=EDTFI1.
REWRITEVISFL1.
FTN(OPT=1,1=VISFL1).
ATTACH,LRCGOSF/UN=LIBRARY,NA.
LDSET,LIB=LRCGOSF,PRESET=ZERO.
LOG.

PLOT,VARIA.
REWRITE,SOLNXX.
COPY8R,SOLNXX,FILETC.
REWRITE,FILETC.
SAVE,FILETC.

7/8/9
IDENT DUM1
*INSERT CORD1.3
C**
7/8/9
RSI1,HXLL,/, 30/;
RSI1,HXLL,/, 23/;
RSI1,HXLL,/, 23/;
END

7/8/9
INPUT DATA, SEE EXPLANATION ABOVE.

6/7/8/9

REWRITEVISFL1.
NOTE: AFTER FIRST 7/8/9 CARD 3 CARDS ARE INPUT TO
Satisfy input for update utility.

AFTER 2ND 7/8/9 CARD EDIT CARDS ARE INPUT. THE
PROGRAM IS WRITTEN SO THAT MAXIMUM NUMBER OF
POINTS IN X, Y, Z DIRECTION CAN BE CHANGED BY SIMPLE INPUT. IN THE ABOVE CASE FIELD SIZE OF 30, 23, 23 IS ASSUMED. IN THE THREE CARDS AFTER RSII AND MXLL I, J, K SHOULD BE INPUT RESPECTIVELY. THIS COULD NOT BE SHOWN ABOVE BECAUSE TEXT EDITOR CHANGES TEXT IN COMMENTS CARDS ALSO.

IT IS ASSUMED THAT PROGRAM RESIDES ON FILE KMRCD1 UNDER USER NO. OF 375732N AND USER 479019C HAS PERMISSION TO USE IT. FURTHERMORE, IT IS ALSO ASSUMED THAT THE COORDINATE SYSTEM SOLUTION IS TO BE STORED ON FILE FILETC. THIS NAME CAN BE CHANGED BY CHANGING THREE CARDS TOWARDS THE END OF THE RUNSTREAM.

THE PLOTS ON CALCOMP CAN BE OBTAINED BY REPLACING PLOT, VARIAN CARD BY FOLLOWING CARDS.

COMMON /BBBB/ Y(JMXXL, KMXLL), Z(JMXXL, KMXXL), O(JMXXL, KKMXXL), R(JMXXL, KMXLL),
1 X(JMXXL, KMXLL), I1(IIMXXL), I2(IIMXXL),
2 COMMON /AAAA/ LIMIT, JLIMIT, KLIMIT, JMAX, JMAX, KMAX, JMAX, IMAX, H1, H2,
3 IPRT1, IPRT2, ITERMX, XI, XR, RX, R3, HEDI(4), HEDI(4),
4 COMMON /BBBB/ YERRMX, ZERRMX, IL, IYLOCJ, IYLOCK, IZLOCJ, IZLOCK, /0, 2*0, 5*0/

DATA KKK, YERRMX, ZERRMX, IL, IYLOCJ, IYLOCK, IZLOCJ, IZLOCK /0, 2*0, 5*0/

LIMIT = JMXXL
JLIMIT = JMXXL
KLIMIT = KMXXL

INITIALIZE Q, R, Y AND Z TO ZERO.

DO 10 J = 1, JLIMIT
10 CONTINUE
Q(J) = 0.0

DO 20 K = 1, KLIMIT
20 CONTINUE
R(K) = 0.0

DO 30 J = 1, JLIMIT
30 CONTINUE
Y(J, K) = 0.0
CREATION RUN

CARDS ENCOUNTERED IN INPUT

UPDATE 1.2-452

79/06/26  13.30.37.  PAGE  5

```
Z(J,K) = 0.0
30 CONTINUE
C**
C** READ INPUT
C**
READ(5,1000) JMAX,KMAX,IPRT1,IPRT2,IPLT1,IPLT2,ITERMX,R1,R2,R3
IF(JMAX.GT.JLIMIT.OR.KMAX.GT.KLIMIT) GO TO 50
GO TO 60
C**
50 WRITE(6,1020) JMAX,KMAX,JLIMIT,KLIMIT
CALL EXIT
C**
60 CONTINUE
READ(5,1030) (Y(J,K),Z(J,K),J=1,JMAX)
READ(5,1030) (Y(J,K),Z(J,K),K=1,KMAX)
C**
READ IN POINTS ALONG J = 1, K = 1 LINE (X-DIRECTION)
C**
READ(5,1000) IMAX
IF(IMAX.LE.ILIMIT) GO TO 65
WRITE(6,1090) IMAX,ILIMIT
STOP
65 CONTINUE
READ(5,1100) (X(I),Y(I),Z(I),I=1,IMAX)
C**
SPECIFY INITIAL GUESS ON THE WHOLE FIELD
C**
DO 70 J = 2,JMAX
   Z(J,KMAX) = Z(J,KMAX)
70 CONTINUE
DO 80 K = 2,KMAX
   Y(JMAX,K) = Y(JMAX,1)
80 CONTINUE
C**
J1 = JMAX - 1
K1 = KMAX - 1
DO 100 J = 2,J1
   DO 100 K = 2,K1
      Y(J,K) = (Y(J,KMAX) - Y(J,1)) * ((Y(J,1) - Y(1,1)) / 
          (Y(J,KMAX) - Y(1,K)) + Y(1,K))
      Z(J,K) = (Z(J,KMAX) - Z(J,1)) * ((Z(J,1) - Z(1,1)) / 
           (Z(J,KMAX) - Z(1,K)) + Z(1,K))
      1
100 CONTINUE
```
DO 102 J = 2, J1
   Y(J,KMAX) = Y(J,KMAX-1)
102 CONTINUE
      DO 104 K = 2, K1
         Z(JMAX,K) = Z(JMAX-1,K)
104 CONTINUE
C++      READ PRINT, PLOT HEADINGS. TWO CARDS ARE READ IN.
C++      CALL PLOT INITIALIZING ROUTINE PSEUDO IF PLOT OPTIONS
C++      SET.
C++
C++      READ(5,1080) (HE01(I),I=1,4)
C++      READ(5,1080) (HE02(I),I=1,4)
C++      IF(IPLT1.EQ.0.AND.IPLT2.EQ.0) GO TO 110
C++              CALL PSEUDO
C++              CALL FONTS(1)
C++
C++      SEE IF INITIAL GUESS NEEDS TO BE PRINTED
C++
C++      IF(IPRT1.EQ.1) CALL IPRTC(1,KKK,YERRMX,ZERRMX,  
C++      IYLOCJ,1YLOCK,  
C++              IZLOCJ,IZLOCK)
C++
C++      SEE IF INITIAL GUESS NEED TO BE PLOTTED
C++
C++      IF(IPLTL.EQ.1) CALL IPLTC(1,KKK,YERRMX,ZERRMX,  
C++      IYLOCJ,1YLOCK,  
C++              IZLOCJ,IZLOCK)
C++
                CONTINUE
C++
                COMPARE Q AND R. FIRST YETA OR ZTAU IS COMPUTED.
C++
                Q = YETAETA / (YETA*YETA+YETA)
C++
                R = ZTAUTAU / (ZTAU*ZTAU+ZTAU)
C++
        DO 120 J = 2, J1
               A1 = 0.5 * (Y(J+1,1) - Y(J-1,1))
               Q(J) = -(Y(J+1,1) - 2.0*Y(J,1) + Y(J-1,1)) / (A1*A1+A1)
120 CONTINUE
C++
        DO 140 K = 2, K1
               A1 = 0.5 * (Z(1,K+1) - Z(1,K-1))
               R(K) = -(Z(1,K+1) - 2.0*Z(1,K) + Z(1,K-1)) / (A1*A1+A1)
140 CONTINUE
C++      START COMPUTING SOLUTION.
CREATION RUN

CARDS ENCOUNTERED IN INPUT

UPDATE 1-2-452

79/06/26  13:30:37  PAGE 7

CORD1 265
CORD1 266
CORD1 267
CORD1 268
CORD1 269
CORD1 270
CORD1 271
CORD1 272
CORD1 273
CORD1 274
CORD1 275
CORD1 276
CORD1 277
CORD1 278
CORD1 279
CORD1 280
CORD1 281
CORD1 282
CORD1 283
CORD1 284
CORD1 285
CORD1 286
CORD1 287
CORD1 288
CORD1 289
CORD1 290
CORD1 291
CORD1 292
CORD1 293
CORD1 294
CORD1 295
CORD1 296
CORD1 297
CORD1 298
CORD1 299
CORD1 300
CORD1 301
CORD1 302
CORD1 303
CORD1 304
CORD1 305
CORD1 306
CORD1 307
CORD1 308

KKK = 0
IERR = 0
DD  500 I1 = 1, ITERMX
YERRMX = 0.0
ZERRMX = 0.0
CALL CALCOR(YERRMX,ZERRMX, IYLOCJ, IYLOCK, IZLOCJ, IZLOCK)

C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++
C++

IERR = IERR + 1
IF(IERR.NE.50) GO TO 160
IERR = 0
IF(IPRT1.EQ.1) CALL IPRTC(Z, KKK, YERRM, ZERRM, IYLOCJ, IYLOCK, IZLOCJ, IZLOCK)

WRITE(6, 1040) II, IYLOCJ, IYLOCK, YERRMX, IZLOCJ, IZLOCK, ZERRMX

IF(YERRMX.GT.R21) GO TO 160
IF(ZERRMX.GT.R31) GO TO 180

SOLUTION CONVERGED, GET OUT OF THE LOOP, KKK = 1
INDICATES SOLUTION CONVERGED

KKK = 1
GO TO 520

160 CONTINUE

CONTINUE

IF(YERRMX.GT.R2) GO TO 160
IF(ZERRMX.GT.R3) GO TO 180

SOLUTION CONVERGED, GET OUT OF THE LOOP, KKK = 1
INDICATES SOLUTION CONVERGED

CONTINUE

500 CONTINUE
520 CONTINUE

SEE IF CONVERGED OR PARTIALLY CONVERGED SOLUTION IS TO
BE PLOTTED AND PRINTED OR NOT

IF(IPRT2.EQ.1) CALL IPRTC(Z, KKK, YERRM, ZERRM, IYLOCJ, IYLOCK, IZLOCJ, IZLOCK)

IF(IPRT2.EQ.1) CALL IPRTC(Z, KKK, YERRM, ZERRM, IYLOCJ, IYLOCK, IZLOCJ, IZLOCK)

STORE CONVERGED SOLUTION

IF(KKK.EQ.0) GO TO 540
CREATION RUN CARDS ENCOUNTERED IN INPUT UPDATE 1.2-452* 79/06/26 13.30.37* PAGE 8

C**
C** SOLUTION DID NOT CONVERGE. WRITE A MESSAGE
C**
540 CONTINUE
WRITE(6,1060) YERRMX, IYLOCJ, IYLOCK, ZERRMX, IZLOCJ, IZLOCK, R2, R3,
1 ITERM X
C**
C** ALL FUNCTIONS COMPLETE
C**
560 CONTINUE
STOP
C**

C** FORMAT STATEMENTS
C**
1000 FORMAT(715,5X,3F10.0)
1020 FORMAT("1"///"5x","** NO OF POINTS IN Y OR Z greater than"," 1 " MAXIMUM ALLOWED="/5x","** POINTS INPUT = ",215,"///"5x"," 2 " MAXIMUM ALLOWED = ",215,"///"5x","** RUN ABORTED")
1030 FORMAT(2F10.0)
1040 FORMAT("0","5x","ITER NO.,IYLOCJ,IYLOCK,YERRMX,IZLOCJ,IZLOCK",""," ",1 "YERRMX = ",315,"E15.6,215,"E15.6)
C**
1060 FORMAT("1"///"5x","** SOLUTION DID NOT CONVERGE. SOLUTION"," 1 " NOT STORED ON UNIT 1"///"5x","** MAX ERROR IN Y = ",E15.6, 2 " OCCURED AT LOCATION = ",15","/5x","** MAX ERROR IN"," 3 "Z = ",E15.6,2 OCCURED AT LOCATION = "///"15","/5x","** MAX ERROR ALLOWED IN Y AND Z = ",15",2E15.6,"///"5x","** NO"," 4 " OF ITERATIONS PERFORMED = ",15")
1080 FORMAT(4A10)
1090 FORMAT("1"///"5x","** NO. OF POINTS IN X-DIRECTION greater than"," 1 " MAXIMUM ALLOWED="/5x",** POINTS INPUT = ",15,"///"5x", 2 " MAXIMUM ALLOWED = ",15,"///"5x","** RUN ABORTED")
1100 FORMAT(3F10.0)
1110 FORMAT(6,15,8A10)
1120 FORMAT(1X,"8E15.8)
C**
C** END
SUBROUTINE CALCOR(YERRMX, ZERRMX, IYLOCJ, IYLOCK, IZLOCJ, IZLOCK)
THIS SUBROUTINE ITERATES FOR COORDINATE SYSTEM SOLUTION.

THE SUBROUTINE IS CALLED FROM COORDC PROGRAM

COMMON /BXXX/ Y(JMXXL,KMXXL),Z(JMXXL,KMXXL),G(JMXXL),R(KMXXL)
1 X(JMXXL),Y(JIMXXL),Z(JMXXL)
COMMON /AXXX/ ILIMIT,JLIMIT,KLIMIT,IMAX,JKMAX,IPRT1,IPRT2,
1 IPLT1,IPLT2,ITERMX,R1,R2,R3,I1,HED1(4),HED2(4)

COMPUTE IN INNER FIELDS

J1 = JMXX - 1
K1 = KMXX - 1
DD 200 J = 2, J1
DD 200 K = 2, K1
YPOP0 = Y(J,K)
YPOP1 = Y(J,K+1)
YPOM1 = Y(J,K-1)
YP1PO = Y(J+1,K)
YM1PO = Y(J-1,K)
YM1M1 = Y(J-1,K-1)
YM1P1 = Y(J-1,K+1)
YP1M1 = Y(J+1,K-1)
YP1P1 = Y(J+1,K+1)
ZPOPO = Z(J,K)
ZPOPI = Z(J,K+1)
ZPOM1 = Z(J,K-1)
ZPIPO = Z(J+1,K)
ZM1PO = Z(J-1,K)
ZM1M1 = Z(J-1,K+1)
ZM1P1 = Z(J-1,K-1)
ZPIPO = Z(J+1,K)
Q0O = Q(J)
RRR = R(K)
ALPHA = 0.25*(YP1PO + YPM1)*2.0 + (ZPOP1 - ZPOM1)**2.0
BETA = 0.25*(YP1PO - YPM1) + (YPOM1 - YPOP1) +
1 (ZPIPO - ZMIPO) + (ZPOP1 - ZPOM1)
GAMMA = 0.25*(YP1PO - YPM1)**2.0 + (ZPIPO - ZMIPO)**2.0
AJCB1 = 0.25*(YP1PO - YPM1) + (ZPOP1 - ZPOM1) -
1 (YP1PO - YPM1)*2.0 + (ZPIPO - ZMIPO)
YY1 = 0.5 + ALPHA + (YP1PO + YPM1) - 0.25*BETA*
1 (YP1PO - YPM1 - YPM1 + YPM1) + 0.5 * GAMMA *
CREATION RUN

CARDS ENCOUNTERED IN INPUT

UPDATE 12-452

79/06/26 13:30:37 PAGE 10

2
(YPOPO1 + YPOPO1) - 0.25 * AJCB1 * AJCB1 * (YPOPO1 - YPOPO1) * 

3
QOQ + (YPOPO1 - YPOPO1)*RRR1) / (ALPHA + GAMMA)

C**

ZZZ1 = (0.5 * ALPBA * (ZP1PO + ZM1PO) - 0.25 * BETA * 

1
(ZP1PO - ZP1PO - ZM1PO + ZM1PO) + 0.5 * GAMMA * (ZPOPO1 + 

2
+ ZPOM1) + 0.25 * AJCB1 * AJCB1 * ((ZP1PO - ZM1PO) * QOQ + 

3
(ZPOPO1 - ZPOPO1)*RRR1) / (ALPHA + GAMMA)

C**

INTERMEDIATE VALUE CALCULATED. COMPUTE NEW VALUE USING

GAUSS SEIDELL ITERATION

YYY1 = R1 * YYY1 + (1.0 - R1) * YPOPO

ZZZ1 = R1 * ZZZ1 + (1.0 - R1) * ZPOPO

FIN MAX ERROR AND ITS LOCATION

ERR1 = ABS(YYY1 - YPOPO)

IF(ERR1 <= YERRMX) GO TO 30

IYLOCJ = J

IYLOCK = K

YERRMX = ERR1

30 CONTINUE

ERR1 = ABS(ZZZ1 - ZPOPO)

IF(ERR1 <= ZERRMX) GO TO 60

IZLOCJ = J

IZLOCK = K

ZERRMX = ERR1

60 CONTINUE

SET NEW VALUES

Y(J,K) = YYY1

Z(J,K) = ZZZ1

200 CONTINUE

APPLY NEUMANN CONDITIONS ON R.H.S. AND UPPER BOUNDARY

DO 220 K = Z,K1

1(ZJMAX,K) = Z(JMAX-1,K)

220 CONTINUE

DO 240 J = 2,J1
SUBROUTINE IPRTC(ISOLN,KKK,YERRM,ZERRM, IYLOCJ,IYLOCK, 
1 IZLOCJ,IZLOCK)
C**
C** THIS SUBROUTINE PRINTS THE COORDINATE SYSTEM IN Y AND Z
C** PLANE. ISOLN = 1 INDICATES INITIAL GUESS IS TO BE PRINTED.
C** ISOLN = 2 INDICATES CONVERGED OR PARTIAL CONVERGED SOLUTION IS TO BE PRINTED. KKK = 0 INDICATES PARTIALLY CONVERGED SOLUTION AND KKK = 1 INDICATES CONVERGED SOLUTION
C**
COMMON /BXXX/ Y(JMXX,KMXX),Z(JMXX,KMXX),O(JMXX,KMXX),R(KMXX),
1 X(IMXX,YIMXX),Z1(IMXX),Z2(IMXX), 
 COMMON /AXXX/ ILIMIT,JLIMIT,KLIMIT,IAXM,JKAX,IPRT1,IPRT2,
1 IPLT1,IPLT2,ITERMX,R1,R2,R3,I1,HED1,4,HED2,4
C**
C** PRINT HEADINGS
C** IF(ISOLN.EQ.1) WRITE(6,1000)
IF(ISOLN.EQ.2.AND.KKK.EQ.0) WRITE(6,1010)
IF(ISOLN.EQ.2.AND.KKK.EQ.1) WRITE(6,1020)
WRITE(6,1025) HED1
WRITE(6,1025) HED2
WRITE(6,1030) JMAX,KMAX
IF(ISOLN.EQ.1) GO TO 50
WRITE(6,1040) I1
WRITE(6,1050) YERRM,ZERRM, IYLOCJ,IYLOCK
WRITE(6,1060) ZERRM,IZLOCJ,IZLOCK
50 CONTINUE
C**
C** PRINT ARRAYS
C** DO 100 J = 1,JMAX
WRITE(6,1070) J
WRITE(6,1080) ((Y(J,K),Z(J,K))K = 1,KMAX)
100 CONTINUE
C** 'PRINT J=1, K=1, I=1,IMAX
C** WRITE(6,1200)
CREATION RUN

CARDS ENCOUNTERED IN INPUT

UPDATE 1.2-452... 79/06/26. 13.30.37... PAGE 12

WRITE(6,1210) ((XI, YI, ZI), I=1, IMAX)

C* FORMAT STATEMENTS
C*
1000 FORMAT("M1/5X,"*** INITIAL SOLUTION ***")
1010 FORMAT("M1/4X,"*** PARTIALLY CONVERGED SOLUTION ***")
1020 FORMAT("M1/4X,"*** CONVERGED SOLUTION ***")
1025 FORMAT("M5/4X>4A10")
1050 FORMAT("M5/5X,JMAX = M, IMAX = N")
1040 FORMAT("M5/4X,MAM MAX ITERATIONS PERFORMED = N")
1050 FORMAT("M5/3X,MAX Y-ERROR = N,E15.6,2X,AT LOC. J,K")
1060 FORMAT("M5/3X,MAX Z-ERROR = N,E15.6,2X,AT LOC. J,K")
1070 FORMAT("M5/3X,J = N")
1080 FORMAT("M5/3X,J = N")
1200 FORMAT("M5/5X,J = N")
1210 FORMAT("M5/5X,J = N")

C* ALL FUNCTIONS COMPLETE
C* RETURN
END

SUBROUTINE IPlTC(ISOLN, KKK, YERRMX, ZERRMX, IYLOCJ, IYLOCK,
1 IZLOCJ, IZLOCK)
C*
THIS SUBROUTINE IS USED TO PLOT INITIAL, PARTIALLY CONVERGED
1 OR CONVERGED SOLN. ISOLN = 1 INDICATES INITIAL GUESS
1 ISOLN = 2 AND KKK = 0 INDICATES PARTIALLY CONVERGED SOLN.
1 ISOLN = 2 AND KKK = 1 INDICATES CONVERGED SOLN.

GLOBAL COMMON STATEMENTS
C*
COMMON /BXXX/ Y(JMXLL), X(JMXLL), Z(JMXLL), O(JMXLL), R(KMXLL),
1 X(JMXLL), Y(IIMXL), Z(IIMXL), O(IIMXL), R(IIMXL)
COMMON /A XXX/ JLIMIT, JLIMIT, JLIMIT, JLIMIT, IMAX, JMAX, KMAX, IPRT1, IPRT2,
1 IPLT1, IPLT2, ITERMX, R1, R2, R3, I1, HED1(4), HED2(4)

LOCAL DIMENSION STATEMENTS
C*
DIMENSION YDOM(KMXLL), YDOM1(JMXLL), ZDOM(KMXLL), ZDOM1(JMXLL),
1 HED(4)
CRD1: 929
CRD1: 530
CRD1: 531
CRD1: 532
CRD1: 533
CRD1: 534
CRD1: 535
CRD1: 536
CRD1: 537
CRD1: 538
CRD1: 539
CRD1: 540
CRD1: 541
CRD1: 542
CRD1: 543
CRD1: 544
CRD1: 545
CRD1: 546
CRD1: 547
CRD1: 548
CRD1: 549
CRD1: 550
CRD1: 551
CRD1: 552
CRD1: 553
CRD1: 554
CRD1: 555
CRD1: 556
CRD1: 557
CRD1: 558
CRD1: 559
CRD1: 560
CRD1: 561
CRD1: 562
CRD1: 563
CRD1: 564
CRD1: 565
CRD1: 566
CRD1: 567
CRD1: 568
CRD1: 569
CRD1: 570
CRD1: 571
CRD1: 572
C**
YMIN = 10000.0
ZMIN = 10000.0
YMAX = -10000.0
ZMAX = -10000.0
DO 150 J = 1, JMAX
Do 150 K = 1, KMAX
IF(Y(J,K) LT YMIN) YMIN = Y(J,K)
IF(Y(J,K) GT YMAX) YMAX = Y(J,K)
IF(Z(J,K) LT ZMIN) ZMIN = Z(J,K)
IF(Z(J,K) GT ZMAX) ZMAX = Z(J,K)
150 CONTINUE
C**
PLT DATA. ZETA LINES FIRST
C**
DO 210 J = 1, JMAX
 Do 200 K = 1, KMAX
 YDUM(K) = Y(J,K)
 ZDUM(K) = Z(J,K)
200 CONTINUE
CALL INFOPLT(C,KMAX,YDUM,1,ZDUM,1,YMIN,YMAX,ZMIN,ZMAX,1
1,0,0,HEO,0,HEO,0,5,0,5,0,0,0,2,0)
210 CONTINUE
C**
PLOT ETA LINES
C**
KK = 0
DO 250 K = 1, KMAX
IF(K EQ KMAX) KK = 1
 Do 240 J = 1, JMAX
 YDUM1(J) = Y(J,K)
 ZDUM1(J) = Z(J,K)
240 CONTINUE
CALL INFOPLT(KK,KMAX,YDUM1,1,ZDUM1,1,YMIN,YMAX,ZMIN,ZMAX,1
1,0,0,HEO,0,HEO,0,5,0,5,0,0,0,2,0)
250 CONTINUE
C**
RETURN
C**
1000 FORMAT(7HJMAX = ,I5,10H KMAX = ,I5)
C**
END
B. **Job Control Cards**

A list of job control cards needed to run this program are included in the comment section of the main program COORDC.

C. **Input Explanation**

The instructions for input preparation are included in the comment section of the main program COORDC.

D. **Sample Input**

Sample input listing follows on the next page.

E. **Sample Output**

The sample output generated by the sample input of section D follows sample input.
## D. Sample Input

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### FORTRAN DATA CODING FORM

**NASA-Langley Form 67 (MAR 69)**

**NOTE:** WRITE NUMBERS 0-9, LETTERS I, U, G ≠ C, SYMBOLS / . *
<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: WRITE NUMBERS 10, LETTERS I O U G Z C, SYMBOLS / . *
### Fortran Statement

<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>CONTINUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
<td>6</td>
</tr>
<tr>
<td>0 1 2 3 4 5 6</td>
<td>7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65</td>
</tr>
<tr>
<td>0.0 1364 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 2048 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 2737 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 3446 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 4212 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 5119 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 6323 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 8052 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 1058 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 1412 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 1766 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 2120 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 2474 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 2828 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 3182 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 3536 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 3890 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 4244 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 4598 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 4952 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 5306 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 5660 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 6014 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 6368 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
<tr>
<td>0.0 6722 0.0 0.3 0.1 1.38 0.0 0.3 0.1 1.38</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** WRITE NUMBERS 10, LETTERS I, U, G, Z, C, SYMBOLS /, *
<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>CONTINUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64</td>
<td></td>
</tr>
<tr>
<td>0. 7 0 7.6 0 0...0 3.0 1.1 3.8 0. 0...0 3.0 1.1 3.8</td>
<td></td>
</tr>
<tr>
<td>0. 7 4 3 0 0 0...0 3.0 1.1 3.8 0. 0...0 3.0 1.1 3.8</td>
<td></td>
</tr>
<tr>
<td>0. 7 7 8 4 0 0...0 3.0 1.1 3.8 0. 0...0 3.0 1.1 3.8</td>
<td></td>
</tr>
<tr>
<td>CORNER, FLOW WITH FILLET</td>
<td></td>
</tr>
<tr>
<td>CORDINATE, SYSTEM; SIZE = 3.0 X 2.3 X 2.3</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: WRITE NUMBERS 10, LETTERS I O U G Z C, SYMBOLS / , *
**INITIAL SOLUTION**

**CORNER FLOW WITH FILLET**

**COORDINATE SYSTEM. SIZE=30X23X23**

<table>
<thead>
<tr>
<th>J = 1</th>
<th>(Y-Z ARRAY)</th>
<th>JMAX = 23</th>
<th>KMAX = 23</th>
</tr>
</thead>
<tbody>
<tr>
<td>J = 1</td>
<td>(Y-Z ARRAY)</td>
<td>JMAX = 23</td>
<td>KMAX = 23</td>
</tr>
<tr>
<td>J = 2</td>
<td>(Y-Z ARRAY)</td>
<td>JMAX = 23</td>
<td>KMAX = 23</td>
</tr>
<tr>
<td>J = 3</td>
<td>(Y-Z ARRAY)</td>
<td>JMAX = 23</td>
<td>KMAX = 23</td>
</tr>
<tr>
<td>J = 4</td>
<td>(Y-Z ARRAY)</td>
<td>JMAX = 23</td>
<td>KMAX = 23</td>
</tr>
<tr>
<td>J = 5</td>
<td>(Y-Z ARRAY)</td>
<td>JMAX = 23</td>
<td>KMAX = 23</td>
</tr>
<tr>
<td>J = 6</td>
<td>(Y-Z ARRAY)</td>
<td>JMAX = 23</td>
<td>KMAX = 23</td>
</tr>
</tbody>
</table>

---

**E. SAMPLE OUTPUT**
<table>
<thead>
<tr>
<th>J</th>
<th>Y-Z ARRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>(Y-Z ARRAY)</td>
</tr>
<tr>
<td></td>
<td>93379E-01</td>
</tr>
<tr>
<td></td>
<td>43390E-03</td>
</tr>
<tr>
<td></td>
<td>90286E-01</td>
</tr>
<tr>
<td></td>
<td>41403E-02</td>
</tr>
<tr>
<td></td>
<td>86358E-01</td>
</tr>
<tr>
<td></td>
<td>94075E-02</td>
</tr>
<tr>
<td></td>
<td>81239E-01</td>
</tr>
<tr>
<td></td>
<td>17482E-01</td>
</tr>
<tr>
<td></td>
<td>75370E-01</td>
</tr>
<tr>
<td></td>
<td>29282E-01</td>
</tr>
<tr>
<td>8</td>
<td>(Y-Z ARRAY)</td>
</tr>
<tr>
<td></td>
<td>11807E-00</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>11510E+00</td>
</tr>
<tr>
<td></td>
<td>37085E-02</td>
</tr>
<tr>
<td></td>
<td>11132E+00</td>
</tr>
<tr>
<td></td>
<td>89788E-02</td>
</tr>
<tr>
<td></td>
<td>10639E+00</td>
</tr>
<tr>
<td></td>
<td>17058E-01</td>
</tr>
<tr>
<td></td>
<td>10074E+00</td>
</tr>
<tr>
<td></td>
<td>28865E-01</td>
</tr>
<tr>
<td>9</td>
<td>(Y-Z ARRAY)</td>
</tr>
<tr>
<td></td>
<td>14734E-00</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>14450E+00</td>
</tr>
<tr>
<td></td>
<td>37085E-02</td>
</tr>
<tr>
<td></td>
<td>16089E+00</td>
</tr>
<tr>
<td></td>
<td>89788E-02</td>
</tr>
<tr>
<td></td>
<td>13619E+00</td>
</tr>
<tr>
<td></td>
<td>17058E-01</td>
</tr>
<tr>
<td></td>
<td>13081E+00</td>
</tr>
<tr>
<td></td>
<td>28865E-01</td>
</tr>
<tr>
<td>10</td>
<td>(Y-Z ARRAY)</td>
</tr>
<tr>
<td></td>
<td>18092E-00</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>17824E+00</td>
</tr>
<tr>
<td></td>
<td>37085E-02</td>
</tr>
<tr>
<td></td>
<td>17403E+00</td>
</tr>
<tr>
<td></td>
<td>89788E-02</td>
</tr>
<tr>
<td></td>
<td>17040E+00</td>
</tr>
<tr>
<td></td>
<td>17058E-01</td>
</tr>
<tr>
<td></td>
<td>16531E+00</td>
</tr>
<tr>
<td>11</td>
<td>(Y-Z ARRAY)</td>
</tr>
<tr>
<td></td>
<td>21042E-00</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>21592E+00</td>
</tr>
<tr>
<td></td>
<td>37085E-02</td>
</tr>
<tr>
<td></td>
<td>21274E+00</td>
</tr>
<tr>
<td></td>
<td>89788E-02</td>
</tr>
<tr>
<td></td>
<td>20859E+00</td>
</tr>
<tr>
<td></td>
<td>17058E-01</td>
</tr>
<tr>
<td></td>
<td>20384E+00</td>
</tr>
<tr>
<td></td>
<td>28865E-01</td>
</tr>
<tr>
<td>J</td>
<td>(Y-Z ARRAY)</td>
</tr>
<tr>
<td>12</td>
<td>25933E-00</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>25701E+00</td>
</tr>
<tr>
<td></td>
<td>37085E-02</td>
</tr>
<tr>
<td></td>
<td>25408E+00</td>
</tr>
<tr>
<td></td>
<td>89788E-02</td>
</tr>
<tr>
<td></td>
<td>25025E+00</td>
</tr>
<tr>
<td></td>
<td>17058E-01</td>
</tr>
<tr>
<td></td>
<td>24587E+00</td>
</tr>
<tr>
<td></td>
<td>28865E-01</td>
</tr>
<tr>
<td>13</td>
<td>(Y-Z ARRAY)</td>
</tr>
<tr>
<td></td>
<td>30303E+00</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30082E+00</td>
</tr>
<tr>
<td></td>
<td>37085E-02</td>
</tr>
<tr>
<td></td>
<td>29825E+00</td>
</tr>
<tr>
<td></td>
<td>89788E-02</td>
</tr>
<tr>
<td></td>
<td>29476E+00</td>
</tr>
<tr>
<td></td>
<td>17058E-01</td>
</tr>
<tr>
<td></td>
<td>29077E+00</td>
</tr>
<tr>
<td></td>
<td>28865E-01</td>
</tr>
<tr>
<td>J = 21 (Y-Z ARRAY)</td>
<td>J = 22 (Y-Z ARRAY)</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>0.6720E+00 0.6716E+00 0.37085E-02 0.67121E+00 0.89788E-02 0.67060E+00 0.17058E-01 0.66991E+00 0.28865E-01</td>
<td></td>
</tr>
<tr>
<td>0.66923E+00 0.66883E+00 0.59911E-01 0.66878E+00 0.91638E-01 0.66878E+00 0.12213E+00 0.66878E+00 0.15711E+00</td>
<td></td>
</tr>
<tr>
<td>0.66878E+00 0.66767E+00 0.2389E+00 0.66767E+00 0.28433E+00 0.66767E+00 0.3208E+00 0.66767E+00 0.36130E+00</td>
<td></td>
</tr>
<tr>
<td>0.66876E+00 0.66767E+00 0.40121E+00 0.66767E+00 0.53049E+00 0.66767E+00 0.57849E+00 0.66767E+00 0.62471E+00</td>
<td></td>
</tr>
<tr>
<td>J = 23 (Y-Z ARRAY)</td>
<td></td>
</tr>
<tr>
<td>0.71209E+00 0.71192E+00 0.37085E-02 0.71169E+00 0.89788E-02 0.71139E+00 0.17058E-01 0.71105E+00 0.28865E-01</td>
<td></td>
</tr>
<tr>
<td>0.71073E+00 0.71051E+00 0.65911E-01 0.71051E+00 0.91638E-01 0.71051E+00 0.12213E+00 0.71051E+00 0.15711E+00</td>
<td></td>
</tr>
<tr>
<td>0.71051E+00 0.71051E+00 0.2388E+00 0.71051E+00 0.28433E+00 0.71051E+00 0.3208E+00 0.71051E+00 0.36130E+00</td>
<td></td>
</tr>
<tr>
<td>0.71051E+00 0.71051E+00 0.40121E+00 0.71051E+00 0.53049E+00 0.71051E+00 0.57849E+00 0.71051E+00 0.62471E+00</td>
<td></td>
</tr>
<tr>
<td>0.75000E+00 0.75000E+00 0.37085E-02 0.75000E+00 0.89788E-02 0.75000E+00 0.17058E-01 0.75000E+00 0.28865E-01</td>
<td></td>
</tr>
<tr>
<td>0.75000E+00 0.75000E+00 0.65911E-01 0.75000E+00 0.91638E-01 0.75000E+00 0.12213E+00 0.75000E+00 0.15711E+00</td>
<td></td>
</tr>
<tr>
<td>0.75000E+00 0.75000E+00 0.2388E+00 0.75000E+00 0.28433E+00 0.75000E+00 0.3208E+00 0.75000E+00 0.36130E+00</td>
<td></td>
</tr>
<tr>
<td>0.75000E+00 0.75000E+00 0.40121E+00 0.75000E+00 0.53049E+00 0.75000E+00 0.57849E+00 0.75000E+00 0.62471E+00</td>
<td></td>
</tr>
</tbody>
</table>

$$\frac{I_{1,S} \cdot I_{1,J} \cdot x_{1}}{I_{1,J} \cdot x_{1}}$$

ITER NO., IYLOC, IYLOCK, YERRM, IZLOC, IZLOCK, ZERRM = 50 13 9 716194E-04 9 13 715067E-04
### PARTIALLY CONVERGED SOLUTION

**CORNER FLOW WITH FILLET**

**COORDINATE SYSTEM**

**SIZE=30X23X23**

<table>
<thead>
<tr>
<th>J = 1 (Y-Z ARRAY)</th>
<th>( J = 2 ) (Y-Z ARRAY)</th>
<th>( J = 3 ) (Y-Z ARRAY)</th>
<th>( J = 4 ) (Y-Z ARRAY)</th>
<th>( J = 5 ) (Y-Z ARRAY)</th>
<th>( J = 6 ) (Y-Z ARRAY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J = 1 ) (Y-Z ARRAY)</td>
<td>( J = 2 ) (Y-Z ARRAY)</td>
<td>( J = 3 ) (Y-Z ARRAY)</td>
<td>( J = 4 ) (Y-Z ARRAY)</td>
<td>( J = 5 ) (Y-Z ARRAY)</td>
<td>( J = 6 ) (Y-Z ARRAY)</td>
</tr>
<tr>
<td>( J = 6 ) (Y-Z ARRAY)</td>
<td>( J = 6 ) (Y-Z ARRAY)</td>
<td>( J = 6 ) (Y-Z ARRAY)</td>
<td>( J = 6 ) (Y-Z ARRAY)</td>
<td>( J = 6 ) (Y-Z ARRAY)</td>
<td>( J = 6 ) (Y-Z ARRAY)</td>
</tr>
<tr>
<td>J = 7 (Y-Z ARRAY)</td>
<td>0.</td>
<td>9337E-01</td>
<td>4393E-01</td>
<td>2981E+00</td>
<td>3445E-01</td>
</tr>
<tr>
<td>-----------------</td>
<td>---</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>J = 6 (Y-Z ARRAY)</td>
<td>0.</td>
<td>8237E-01</td>
<td>5235E-01</td>
<td>7621E-01</td>
<td>1039E+00</td>
</tr>
<tr>
<td>J = 9 (Y-Z ARRAY)</td>
<td>1473E+00</td>
<td>4504E-01</td>
<td>1205E-01</td>
<td>3030E-01</td>
<td>3025E-00</td>
</tr>
<tr>
<td>J = 10 (Y-Z ARRAY)</td>
<td>1809E+00</td>
<td>1737E+00</td>
<td>4410E-01</td>
<td>1705E-01</td>
<td>1669E-01</td>
</tr>
<tr>
<td>J = 11 (Y-Z ARRAY)</td>
<td>2164E+00</td>
<td>4373E-01</td>
<td>2093E-01</td>
<td>1959E-01</td>
<td>1935E-01</td>
</tr>
<tr>
<td>J = 12 (Y-Z ARRAY)</td>
<td>2593E+00</td>
<td>4335E-01</td>
<td>2360E-01</td>
<td>6692E+00</td>
<td>2360E+00</td>
</tr>
<tr>
<td>J = 13 (Y-Z ARRAY)</td>
<td>3030E+00</td>
<td>3489E-01</td>
<td>2962E+00</td>
<td>6393E-01</td>
<td>4342E-01</td>
</tr>
</tbody>
</table>

**Notes:**
- The table contains values in scientific notation.
- The values are likely to represent coordinates or measurements in a 2D or 3D space.
<table>
<thead>
<tr>
<th>J</th>
<th>Y-Z ARRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>(2.871E+00), (1.947E+00), (2.854E+00), (2.379E+00), (2.841E+00), (2.894E+00), (3.325E+00), (2.827E+00), (2.902E+00), (5.321E+00), (2.843E+00), (5.801E+00), (2.820E+00), (6.260E+00)</td>
</tr>
<tr>
<td>15</td>
<td>(3.486E+00), (3.445E+00), (3.434E+00), (3.428E+00), (6.383E-01), (3.409E+00), (8.916E-01), (3.390E+00), (1.194E+00), (3.370E+00), (1.543E+00), (3.316E+00), (3.814E+00)</td>
</tr>
<tr>
<td>16</td>
<td>(4.405E+00), (4.409E+00), (4.342E+00), (4.397E+00), (6.376E-01), (4.387E+00), (8.905E-01), (4.371E+00), (1.191E+00), (4.357E+00), (1.540E+00), (4.337E+00), (3.806E+00)</td>
</tr>
<tr>
<td>17</td>
<td>(4.920E+00), (4.894E+00), (4.342E-01), (4.851E+00), (6.375E-01), (4.874E+00), (8.890E-01), (4.867E+00), (1.191E+00), (4.852E+00), (1.539E+00), (4.831E+00), (3.803E+00)</td>
</tr>
<tr>
<td>18</td>
<td>(5.330E+00), (5.373E+00), (4.342E-01), (5.365E+00), (6.375E-01), (5.397E+00), (6.667E+00), (5.349E+00), (1.190E+00), (5.342E+00), (1.539E+00), (5.324E+00), (3.802E+00)</td>
</tr>
<tr>
<td>19</td>
<td>(5.853E+00), (5.839E+00), (4.342E-01), (5.833E+00), (6.375E-01), (5.827E+00), (8.897E+00), (5.820E+00), (1.190E+00), (5.814E+00), (1.539E+00), (5.799E+00), (3.802E+00)</td>
</tr>
<tr>
<td>20</td>
<td>(6.297E+00), (6.269E+00), (4.342E-01), (6.296E+00), (6.375E-01), (6.276E+00), (8.622E+00), (6.292E+00), (1.190E+00), (6.274E+00), (1.539E+00), (6.277E+00), (3.802E+00)</td>
</tr>
<tr>
<td>J</td>
<td>Y-Z ARRAY</td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
</tr>
<tr>
<td>21</td>
<td>71209E+00</td>
</tr>
<tr>
<td>22</td>
<td>71179E+00</td>
</tr>
<tr>
<td>23</td>
<td>75000E+00</td>
</tr>
<tr>
<td>24</td>
<td>75000E+00</td>
</tr>
<tr>
<td>25</td>
<td>75000E+00</td>
</tr>
<tr>
<td>26</td>
<td>75000E+00</td>
</tr>
<tr>
<td>27</td>
<td>75000E+00</td>
</tr>
<tr>
<td>28</td>
<td>75000E+00</td>
</tr>
<tr>
<td>29</td>
<td>75000E+00</td>
</tr>
<tr>
<td>30</td>
<td>75000E+00</td>
</tr>
</tbody>
</table>

**Table:**

<table>
<thead>
<tr>
<th>J</th>
<th>Y-Z ARRAY</th>
<th>ITER NO., IYLOCJ, IYLOCK, YERRMX, IZLOCJ, IZLOCK, ZERRMX</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>71209E+00</td>
<td>0.</td>
</tr>
<tr>
<td>22</td>
<td>71179E+00</td>
<td>0.</td>
</tr>
<tr>
<td>23</td>
<td>75000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>24</td>
<td>75000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>25</td>
<td>75000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>26</td>
<td>75000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>27</td>
<td>75000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>28</td>
<td>75000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>29</td>
<td>75000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>30</td>
<td>75000E+00</td>
<td>0.</td>
</tr>
</tbody>
</table>

Iter No., IYLOCJ, IYLOCK, YERRMX, IZLOCJ, IZLOCK, ZERRMX = 100 15 10 166944E-04 10 15.166901E-04
### CONVERGED SOLUTION

**CORNER FLOW WITH FILLET**

**COORDINATE SYSTEM:** SIZE = 30X23X23

<table>
<thead>
<tr>
<th>J</th>
<th>1</th>
<th>Y-Z ARRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>2</td>
<td>Y-Z ARRAY</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>Y-Z ARRAY</td>
</tr>
<tr>
<td>J</td>
<td>4</td>
<td>Y-Z ARRAY</td>
</tr>
<tr>
<td>J</td>
<td>5</td>
<td>Y-Z ARRAY</td>
</tr>
<tr>
<td>J</td>
<td>6</td>
<td>Y-Z ARRAY</td>
</tr>
</tbody>
</table>

**JMAX = 23**  
**KMAX = 23**  
**MAX ITERATIONS PERFORMED = 119**

**MAX Y-ERROR:**  
- AT LOC. J, K = 15, 10
- val: 998592E-05

**MAX Z-ERROR:**  
- AT LOC. J, K = 10, 15
- val: 999905E-05

**J**  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23 

**Y-Z ARRAY**

<table>
<thead>
<tr>
<th>J</th>
<th>Y-Z ARRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
</tr>
<tr>
<td>J</td>
<td>2</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
</tr>
<tr>
<td>J</td>
<td>4</td>
</tr>
<tr>
<td>J</td>
<td>5</td>
</tr>
<tr>
<td>J</td>
<td>6</td>
</tr>
<tr>
<td>J</td>
<td>7</td>
</tr>
<tr>
<td>J</td>
<td>8</td>
</tr>
<tr>
<td>J</td>
<td>9</td>
</tr>
<tr>
<td>J</td>
<td>10</td>
</tr>
<tr>
<td>J</td>
<td>11</td>
</tr>
<tr>
<td>J</td>
<td>12</td>
</tr>
<tr>
<td>J</td>
<td>13</td>
</tr>
<tr>
<td>J</td>
<td>14</td>
</tr>
<tr>
<td>J</td>
<td>15</td>
</tr>
<tr>
<td>J</td>
<td>16</td>
</tr>
<tr>
<td>J</td>
<td>17</td>
</tr>
<tr>
<td>J</td>
<td>18</td>
</tr>
<tr>
<td>J</td>
<td>19</td>
</tr>
<tr>
<td>J</td>
<td>20</td>
</tr>
<tr>
<td>J</td>
<td>21</td>
</tr>
<tr>
<td>J</td>
<td>22</td>
</tr>
<tr>
<td>J</td>
<td>23</td>
</tr>
<tr>
<td>J</td>
<td>24</td>
</tr>
<tr>
<td>J</td>
<td>25</td>
</tr>
<tr>
<td>J</td>
<td>26</td>
</tr>
<tr>
<td>J</td>
<td>27</td>
</tr>
<tr>
<td>J</td>
<td>28</td>
</tr>
<tr>
<td>J</td>
<td>29</td>
</tr>
<tr>
<td>J</td>
<td>30</td>
</tr>
<tr>
<td>J</td>
<td>31</td>
</tr>
<tr>
<td>J</td>
<td>32</td>
</tr>
<tr>
<td>J</td>
<td>33</td>
</tr>
<tr>
<td>J</td>
<td>34</td>
</tr>
<tr>
<td>J</td>
<td>35</td>
</tr>
<tr>
<td>J</td>
<td>36</td>
</tr>
<tr>
<td>J</td>
<td>37</td>
</tr>
<tr>
<td>J</td>
<td>38</td>
</tr>
<tr>
<td>J</td>
<td>39</td>
</tr>
<tr>
<td>J</td>
<td>40</td>
</tr>
<tr>
<td>J</td>
<td>41</td>
</tr>
<tr>
<td>J</td>
<td>42</td>
</tr>
<tr>
<td>J</td>
<td>43</td>
</tr>
<tr>
<td>J</td>
<td>44</td>
</tr>
<tr>
<td>J</td>
<td>45</td>
</tr>
<tr>
<td>J</td>
<td>46</td>
</tr>
<tr>
<td>J</td>
<td>47</td>
</tr>
<tr>
<td>J</td>
<td>48</td>
</tr>
<tr>
<td>J</td>
<td>49</td>
</tr>
<tr>
<td>J</td>
<td>50</td>
</tr>
<tr>
<td>J</td>
<td>51</td>
</tr>
<tr>
<td>J</td>
<td>52</td>
</tr>
<tr>
<td>J</td>
<td>53</td>
</tr>
<tr>
<td>J</td>
<td>54</td>
</tr>
<tr>
<td>J</td>
<td>55</td>
</tr>
<tr>
<td>J</td>
<td>56</td>
</tr>
<tr>
<td>J</td>
<td>57</td>
</tr>
<tr>
<td>J</td>
<td>58</td>
</tr>
<tr>
<td>J</td>
<td>59</td>
</tr>
<tr>
<td>J</td>
<td>60</td>
</tr>
<tr>
<td>J</td>
<td>61</td>
</tr>
<tr>
<td>J</td>
<td>62</td>
</tr>
<tr>
<td>J</td>
<td>63</td>
</tr>
<tr>
<td>J</td>
<td>64</td>
</tr>
<tr>
<td>J</td>
<td>65</td>
</tr>
<tr>
<td>J</td>
<td>66</td>
</tr>
<tr>
<td>J</td>
<td>67</td>
</tr>
<tr>
<td>J</td>
<td>68</td>
</tr>
<tr>
<td>J</td>
<td>69</td>
</tr>
<tr>
<td>J</td>
<td>70</td>
</tr>
<tr>
<td>J</td>
<td>71</td>
</tr>
<tr>
<td>J</td>
<td>72</td>
</tr>
<tr>
<td>J</td>
<td>73</td>
</tr>
<tr>
<td>J</td>
<td>74</td>
</tr>
<tr>
<td>J</td>
<td>75</td>
</tr>
<tr>
<td>J</td>
<td>76</td>
</tr>
<tr>
<td>J</td>
<td>77</td>
</tr>
<tr>
<td>J</td>
<td>78</td>
</tr>
<tr>
<td>J</td>
<td>79</td>
</tr>
<tr>
<td>J</td>
<td>80</td>
</tr>
<tr>
<td>J</td>
<td>81</td>
</tr>
<tr>
<td>J</td>
<td>82</td>
</tr>
<tr>
<td>J</td>
<td>83</td>
</tr>
<tr>
<td>J</td>
<td>84</td>
</tr>
<tr>
<td>J</td>
<td>85</td>
</tr>
<tr>
<td>J</td>
<td>86</td>
</tr>
<tr>
<td>J</td>
<td>87</td>
</tr>
<tr>
<td>J</td>
<td>88</td>
</tr>
<tr>
<td>J</td>
<td>89</td>
</tr>
<tr>
<td>J</td>
<td>90</td>
</tr>
<tr>
<td>J</td>
<td>91</td>
</tr>
<tr>
<td>J</td>
<td>92</td>
</tr>
<tr>
<td>J</td>
<td>93</td>
</tr>
<tr>
<td>J</td>
<td>94</td>
</tr>
<tr>
<td>J</td>
<td>95</td>
</tr>
<tr>
<td>J</td>
<td>96</td>
</tr>
<tr>
<td>J</td>
<td>97</td>
</tr>
<tr>
<td>J</td>
<td>98</td>
</tr>
<tr>
<td>J</td>
<td>99</td>
</tr>
</tbody>
</table>

**Z**  
1  
2  
3 

**J**  
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  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
<table>
<thead>
<tr>
<th>Y-Z ARRAY</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.31793E-01</td>
<td>4.33955E-01</td>
<td>4.33655E-01</td>
<td>4.23975E-01</td>
<td>4.2941E-01</td>
<td>4.3450E-01</td>
<td>4.3416E-01</td>
<td>3.9367E+00</td>
<td>2.9442E+01</td>
<td>2.4201E+00</td>
<td>2.4396E+00</td>
<td>2.4401E+00</td>
<td>2.4396E+00</td>
</tr>
<tr>
<td>5</td>
<td>4.2549E+00</td>
<td>4.2505E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
<td>4.2549E+00</td>
</tr>
<tr>
<td>9</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
<td>4.2994E+00</td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</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>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>1.0</td>
<td>2.0</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>1.0</td>
<td>2.0</td>
<td>3.0</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>2.0</td>
<td>3.0</td>
<td>4.0</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>3.0</td>
<td>4.0</td>
<td>5.0</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>4.0</td>
<td>5.0</td>
<td>6.0</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>5.0</td>
<td>6.0</td>
<td>7.0</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>6.0</td>
<td>7.0</td>
<td>8.0</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>7.0</td>
<td>8.0</td>
<td>9.0</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>8.0</td>
<td>9.0</td>
<td>10.0</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>9.0</td>
<td>10.0</td>
<td>11.0</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>10.0</td>
<td>11.0</td>
<td>12.0</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: The table represents coordinates in a 3D space, with X, Y, and Z values.*
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.62749E+00</td>
<td>0.19234E+00</td>
<td>0.62710E+00</td>
<td>0.23515E+00</td>
<td>0.62642E+00</td>
<td>0.32955E+00</td>
</tr>
<tr>
<td>0.62592E+00</td>
<td>0.43029E+00</td>
<td>0.62477E+00</td>
<td>0.80095E+00</td>
<td>0.62582E+00</td>
<td>0.53076E+00</td>
</tr>
<tr>
<td>0.62544E+00</td>
<td>0.66945E+00</td>
<td>0.62542E+00</td>
<td>0.71091E+00</td>
<td>0.62542E+00</td>
<td>0.75000E+00</td>
</tr>
<tr>
<td>J = 21</td>
<td>Y-Z ARRAY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.67204E+00</td>
<td>0.67169E+00</td>
<td>0.6706E+00</td>
<td>0.66973E+00</td>
<td>0.66943E+00</td>
<td>0.66943E+00</td>
</tr>
<tr>
<td>0.71209E+00</td>
<td>0.71193E+00</td>
<td>0.71147E+00</td>
<td>0.71120E+00</td>
<td>0.71090E+00</td>
<td>0.71090E+00</td>
</tr>
<tr>
<td>0.75000E+00</td>
<td>0.75000E+00</td>
<td>0.75000E+00</td>
<td>0.75000E+00</td>
<td>0.75000E+00</td>
<td>0.75000E+00</td>
</tr>
<tr>
<td>J = 23</td>
<td>Y-Z ARRAY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.70000E+00</td>
<td>0.70000E+00</td>
<td>0.70000E+00</td>
<td>0.70000E+00</td>
<td>0.70000E+00</td>
<td>0.70000E+00</td>
</tr>
<tr>
<td>J=1,K=1,X=1,IMAX (X,Y,Z ARRAY)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
</tr>
<tr>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
</tr>
<tr>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
</tr>
<tr>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
</tr>
<tr>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
<td>0.30114E-01</td>
</tr>
</tbody>
</table>
THE PLOT CONTROL CARD IMAGE IS:

PLOT, VARIAN

<table>
<thead>
<tr>
<th>FRAME</th>
<th>XO</th>
<th>YO</th>
<th>XM</th>
<th>YM</th>
<th>CAL. POS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.000000E+00</td>
<td>1.000000E+00</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1.000000E+00</td>
<td>1.000000E+00</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1.000000E+00</td>
<td>1.000000E+00</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1.000000E+00</td>
<td>1.000000E+00</td>
<td>0</td>
</tr>
</tbody>
</table>
CORNER FLOW WITH FILLET
COORDINATE SYSTEM. SIZE 30X23X23
--- INITIAL GUESS ---
JMAX 23  KMAX 23
CORNER FLOW WITH FILLET
COORDINATE SYSTEM, SIZE 30X23X23

CONVERGED SOLUTION
JMAX 23  KMAX 23
III. Programmer/Analyst Section

The program is written such that the array size in $\xi$, $\eta$, $\zeta$ direction can be varied from problem to problem by the use of test editor on CDC 6600 machine. The array sizes are coded to be IMXLL, JMXLL, KMXLL in $\xi$, $\eta$ and $\zeta$ direction respectively. These are changed to the appropriate numeric number by the test editor prior to compiling the program for execution. The job control cards, input explanation and other helpful hints are provided towards the beginning of main program COORDC. The program is extensively commented. Though the computer code is not structured in the true sense, extreme care was taken to avoid branching back in the code unless absolutely necessary. A brief discussion of each subroutine and their function follow.

A. Main Program COORDC

This program performs the following function,

i) Initialize variables to 0.0.

ii) Read in input data. If the field size is greater than the maximum field size allowed then the program aborts with a diagnostic message. The maximum field size allowed is ILIMIT, JLIMIT, KLIMIT and the field size input is IMAX, JMAX, KMAX. ILIMIT, JLIMIT, and KLIMIT are set to numerical value by the text editor. The storage requirement in N-S code require that IMAX, JMAX, KMAX be equal to ILIMIT, JLIMIT, KLIMIT respectively.

iii) Specify initial guess.

iv) Check to see if initial guess is to be printed, plotted or not. If so, then the appropriate subroutines are called.

v) Compute attraction parameter $Q$ and $R$. 
vi) Start computing solution. Do loop for maximum iterations (ITERMX) is set. For each iteration subroutine CALCOR is called. CALCOR computes solution for each iteration and also passes back maximum error and its location.

vii) Print maximum error and partially converged solution every 50 iterations. Partially converged solution is printed only if the flag to print the initial guess is set.

viii) Check maximum error in y and z for convergence. If convergence is reached, then an exit from the DO loop is taken.

ix) After ITERMX iterations or convergence, which ever comes first, some more print/plot and storing of data is done.

x) Check to see if partially or converged solution is to be printed/plotted.

xi) If convergence has occurred, then store solution on unit 1. If convergence has not occurred, then the maximum errors in y and z computation at the last iteration and their location are printed out.

B. Subroutine CALCOR

This subroutine advances the solution 1 iteration. The maximum errors and their location are computed and passed back to the calling program COORDC through the argument list. The argument list is,

YERRMX - maximum error in y
ZERRMX - maximum error in z
IYLOCJ - maximum y-error J location
IYLOCK - maximum y-error K location
IZLOCJ - maximum z-error J location
IZLOCK - maximum z-error K location
C. Subroutine IPRTC

This subroutine is used to print initial guess, partially converged or converged solution. The argument list is,

\[ ISOLN = 1 \text{ indicates initial guess} \]
\[ = 2 \text{ indicates partially or fully converged solution} \]
\[ KKK = \text{applicable only if ISOLN} = 2 \]
\[ = 0 \text{ indicates partially converged solution} \]
\[ = 1 \text{ indicates fully converged solution.} \]

\[ YERRMX, ZERRMX, IYLOCJ, IYLOCK, IZLOCJ, IZLOCK = \text{same as in CALCOR.} \]

D. Subroutine IPLTC

This subroutine is used to plot initial guess, partially converged solution or converged solution. The argument list is same as IPRTC.

E. Block Diagram

A Block Diagram is presented on the next page.
START

INITIALIZE ARRAYS

READ INPUT CARDS AND CHECK FOR FIELD SIZE

FIELD SIZE PROPER

WRITE ERROR MESSAGE

STOP

COMPUTE INITIAL GUESS

PRINT/PLT INITIAL GUESS IF IPI, IPRT\_1, IPRT\_2 FLAGS SET BY CALLING IPLTC & IPRTC

COMPUTE ATTRACTION COEFFICIENTS Q & R
1

SET

KKK = 0

DO LOOP FOR

ITERMX

ITERATIONS

CALL CALCOR

to compute
solutions &
max. error

PRINT MAX

ERROR & PARTIALLY

CONVERGED SOLN

EVERY 50 ITERATIONS

DO LOOP

LIMIT OVER

NO

MAX ERROR

< MAX ALLOWED

YES

SET

KKK = 1

2
PRINT/PLOT CONVERGED OR PARTIALLY CONVERGED SOLUTIONS IF FLAGS SET

KKK = 1

NO

WRITE MESSAGE ABOUT SOLN. NOT STORED ON UNIT 1

STOP
The computer program COORDC generates a body-fitted curvilinear coordinate system for corner geometry with or without corner fillets. It is assumed that at any given $\xi$, $x$ remains constant; consequently the only variation is in $y$ and $z$. It is also assumed that for all $\xi$'s in the physical plane the coordinate system in $y$-$z$ plane is similar. This enables solution of coordinate system for one particular $\xi = 1$ ($x$ for $\xi = 1$ is arbitrarily chosen to be 0.0) and the solution for all other $\xi$ plane can be easily specified once the coordinates in the physical plane on the line $1 \leq \xi \leq \text{IMAX}$, $\eta = 1$, $\zeta = 1$ are specified.
End of Document