SECONDARY FLOW IN A CURVED TUBE

by

Donald Greenspan

Appendix: FORTRAN Program for Secondary Flow
by A. B. Schubert

Technical Report #155

June 1972
SECONDARY FLOW IN A CURVED TUBE

by

Donald Greenspan

Appendix: FORTRAN Program for Secondary Flow

by

A. B. Schubert

ABSTRACT

The work of Dean and that of McConalogue and Srivastava on the steady motion of an incompressible fluid through a curved tube of circular cross section is extended through the entire range of Reynolds numbers for which the flow is laminar. The coupled, nonlinear system of partial differential equations which defines the motion is solved numerically by finite differences. Computer calculations are described and physical implications are discussed.
1. Introduction

The flow of a fluid in a curved tube has been of broad interest both experimentally (see, e.g., refs. [3],[8]) and theoretically (see, e.g., [1],[2],[6]). In this paper we will study, in particular, the steady secondary flow of an incompressible fluid through a pipe of circular cross section which is coiled in a circle. Our approach will be numerical and will be applied to the particular model studied qualitatively by Dean [1] and numerically by McConalogue and Srivastava [6]. The method to be used will be a finite difference technique ([4],[5]) and will be both simpler and more comprehensive than that of McConalogue and Srivastava.

Mathematically, the problem to be considered is formulated as follows. Consider a pipe of circular cross-section, coiled in the form of a circle. As shown in Figure 1, let the axis of the circle in which the pipe is coiled be OY and let C be the center of the section of the pipe by a plane through OY which makes an angle \( \theta \) with a fixed axial plane. Let OC be of length L, and let the radius of the cross section be \( a \). The coordinates of any point P of the cross section are denoted by orthogonal coordinates \( (r',a,\theta) \), where \( r' \) is the distance CP and \( \alpha \) is the angle CP makes with OC. Let the velocity components at
P be (U, V, W), where U is in the direction CP, V is perpendicular to U and in the plane of the cross section, and W is perpendicular to this plane. The motion of the fluid is assumed due to a fall in pressure in the direction of increasing θ. It is assumed also that \( \frac{a}{L} \) is relatively small [6]; that U, V, W are independent of θ; and that the motion is steady. Setting

\[
(1.1) \quad r' U = \frac{\partial f}{\partial \alpha}, \quad V = -\frac{\partial f}{\partial r'},
\]

where \( f \), the stream function of the secondary flow, is a function only of \( r' \) and \( \alpha \); defining the constant \( D \) by

\[
(1.2) \quad D = 4R \sqrt{\frac{2a}{L}},
\]

where \( R \) is a given Reynolds number; and introducing the nondimensional variables

\[
(1.3) \quad f = v \phi, \quad W = w \left( \frac{v^2 L}{2a^3} \right)^{1/2}, \quad r' = a r,
\]

where \( v \) is the kinematic viscosity, yields the following equations of motion [6]:

\[
(1.4) \quad \nabla_1^2 w + D = \frac{1}{r} \left( \frac{\partial \phi}{\partial \alpha} \frac{\partial w}{\partial r} - \frac{\partial \phi}{\partial r} \frac{\partial w}{\partial \alpha} \right)
\]

\[
(1.5) \quad -\nabla_1^4 \phi = \frac{1}{r} \left( \frac{\partial \phi}{\partial \alpha} \frac{\partial}{\partial r} - \frac{\partial \phi}{\partial r} \frac{\partial}{\partial \alpha} \right) \nabla_1^2 \phi + w \left( \frac{\partial w}{\partial \alpha} \sin \alpha + \frac{\partial w}{\partial \alpha} \cos \alpha \right) \frac{\partial}{\partial r},
\]
in which

\[ \nabla_1^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \alpha^2} \]

The boundary constraints at \( r = 1 \) are

\[ w = \phi = \frac{\partial \phi}{\partial r} = 0. \]

The problem, then, is to solve the coupled, nonlinear, partial differential equations (1.4) and (1.5) subject to boundary conditions (1.7).

Physically, the experiments of Eustice [3] and Taylor [8] have shown that, for curved tubes, flow can be laminar for much greater Reynolds numbers than is the case for a straight tube, and since Taylor [8] showed that the critical Reynolds number rose for about 5000 for the case \( \frac{L}{a} = 31.9 \), interest has centered on the following range of \( D \):

\[ 0 \leq D \leq 5000. \]

Thus far, convergent results have been obtained only by Dean [1] for \( 0 \leq D \leq 96 \) and by McConalogue and Srivastava [6] for \( 96 \leq D \leq 605.72 \).

In our development of a numerical method which will be convergent for the entire range (1.8), we will be motivated by the powerful difference methods and supportive theory which exist for second order elliptic equations [4]. For this reason, let us rewrite (1.4) and (1.5)
as the following system of second order equations:

\[(1.9) \quad \frac{\partial^2 \phi}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 \phi}{\partial \alpha^2} + \frac{1}{r} \frac{\partial \phi}{\partial r} = -\Omega \]

\[(1.10) \quad \frac{\partial^2 w}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \alpha^2} + \frac{1}{r} \left[ \left( \frac{\partial \phi}{\partial r} \frac{\partial w}{\partial \alpha} \right) + \left( 1 - \frac{\partial \phi}{\partial \alpha} \right) \frac{\partial w}{\partial r} \right] = -D \]

\[(1.11) \quad \frac{\partial^2 \Omega}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 \Omega}{\partial \alpha^2} + \frac{1}{r} \left[ \frac{\partial \phi}{\partial r} \frac{\partial \Omega}{\partial \alpha} + \left( 1 - \frac{\partial \phi}{\partial \alpha} \right) \frac{\partial \Omega}{\partial r} \right] = w \left( \sin \alpha \frac{\partial w}{\partial r} + \frac{\cos \alpha}{r} \frac{\partial w}{\partial \alpha} \right) \]

Observe that \((1.9) - (1.11)\) are, in fact, valid only for \(r > 0\).

The singularity at \(r = 0\) is, nevertheless, not physical, but geometric, and is due to the recasting of the respective equations

\[(1.9') \quad \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -\Omega \]

\[(1.10') \quad \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \left( \frac{\partial \phi}{\partial x} \frac{\partial w}{\partial y} - \frac{\partial \phi}{\partial y} \frac{\partial w}{\partial x} \right) = -D \]

\[(1.11') \quad \frac{\partial^2 \Omega}{\partial x^2} + \frac{\partial^2 \Omega}{\partial y^2} + \left( \frac{\partial \phi}{\partial x} \frac{\partial \Omega}{\partial y} - \frac{\partial \phi}{\partial y} \frac{\partial \Omega}{\partial x} \right) = w \frac{\partial w}{\partial y} \]

into polar coordinates.

But \((1.9') - (1.11')\) yield, readily, the symmetry relationships

\[(1.12) \quad \phi(x, y) = -\phi(x, -y) \]

\[(1.13) \quad \Omega(x, y) = -\Omega(x, -y) \]

\[(1.14) \quad w(x, y) = w(x, -y), \]
which, in turn, will allow us to study our problem on the semicircle defined by \(0 \leq r \leq 1, \ 0 \leq \alpha \leq \pi\). Indeed, from (1.12) and (1.13), one has immediately, in rectangular coordinates, that

\[
(1.15) \quad \phi(x,0) = \Omega(x,0) = 0.
\]

2. Difference Equation Approximations

Fundamental for the method to be developed is the approximation of differential equations (1.9) – (1.11) and (1.10') by difference equations which are associated with diagonally dominant, linear algebraic systems. This will be accomplished by using a combination of central, forward, and backward difference approximations for derivatives as follows, in the same spirit as in [5].

Consider first \(r = 0\) and (1.10'). In rectangular coordinates, and for \(\Delta r > 0\), let the five points \((0,0)\), \((\Delta r,0)\), \((0,\Delta r)\), \((-\Delta r,0)\), \((0,-\Delta r)\) be numbered \(0, 1, 2, 3, 4\), respectively. Then, in the usual subscript notation [5], approximate the second order derivative terms at \((0,0)\) by

\[
(2.1) \quad \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = \frac{-4w_0 + w_1 + w_2 + w_3 + w_4}{(\Delta r)^2}.
\]

Next, set
(2.2) \[ \frac{\partial \phi}{\partial x} = \frac{\phi_1 - \phi_3}{2\Delta r}, \quad \frac{\partial \phi}{\partial y} = \frac{\phi_2 - \phi_4}{2\Delta r} \]

and

(2.3) \[ \epsilon = \phi_1 - \phi_3 \]

(2.4) \[ \beta = \phi_2 - \phi_4 \]

Then, approximate \[ \frac{\partial w}{\partial y} \] and \[ \frac{\partial w}{\partial x} \] by

\[
(2.5) \quad \frac{\partial w}{\partial y} = \begin{cases} 
\frac{w_2 - w_0}{\Delta r}, & \epsilon \geq 0 \\
\frac{w_0 - w_4}{\Delta r}, & \epsilon < 0 
\end{cases}
\]

(2.6) \[ \frac{\partial w}{\partial x} = \begin{cases} 
\frac{w_0 - w_3}{\Delta r}, & \beta \geq 0 \\
\frac{w_1 - w_0}{\Delta r}, & \beta < 0 
\end{cases} \]

If one now defines the quantities \( A, B, C \) by

(2.7) \[ A = -4 - \frac{|\epsilon|}{2} - \frac{|\beta|}{2} \]

(2.8) \[ B = 1 + \frac{|\epsilon|}{2} \]

(2.9) \[ C = 1 + \frac{|\beta|}{2} \]
then the difference approximation of (1.10') which results is

\[
\begin{align*}
Aw_0 + w_1 + Bw_2 + Cw_3 + w_4 &= -(\Delta r)^2 D; \quad \epsilon \geq 0, \quad \beta \geq 0 \\
Aw_0 + Cw_1 + Bw_2 + w_3 + w_4 &= -(\Delta r)^2 D; \quad \epsilon \geq 0, \quad \beta < 0 \\
Aw_0 + w_1 + w_2 + Cw_3 + Bw_4 &= -(\Delta r)^2 D; \quad \epsilon < 0, \quad \beta \geq 0 \\
Aw_0 + Cw_1 + w_2 + w_3 + Bw_4 &= -(\Delta r)^2 D; \quad \epsilon < 0, \quad \beta < 0.
\end{align*}
\]

(2.10)

Consider, next, \( r > 0 \) and (1.9) - (1.11). For given positive values \( \Delta r \) and \( \Delta \alpha \), let the five polar points \( (r, \alpha), (r+\Delta r, \alpha), (r, \alpha+\Delta \alpha), (r-\Delta r, \alpha), (r, \alpha-\Delta \alpha) \) be numbered 0, 1, 2, 3, 4, respectively. Let the second order derivatives in (1.9) - (1.11) be approximated by

\[
\begin{align*}
\left. \frac{\partial^2 \phi}{\partial r^2} \right|_0 &= \frac{\phi_1 - 2\phi_0 + \phi_3}{(\Delta r)^2}, \quad \left. \frac{\partial^2 \phi}{\partial \alpha^2} \right|_0 = \frac{\phi_2 - 2\phi_0 + \phi_4}{(\Delta \alpha)^2} \\
\left. \frac{\partial^2 w}{\partial r^2} \right|_0 &= \frac{w_1 - 2w_0 + w_3}{(\Delta r)^2}, \quad \left. \frac{\partial^2 w}{\partial \alpha^2} \right|_0 = \frac{w_2 - 2w_0 + w_4}{(\Delta \alpha)^2} \\
\left. \frac{\partial^2 \Omega}{\partial r^2} \right|_0 &= \frac{\Omega_1 - 2\Omega_0 + \Omega_3}{(\Delta r)^2}, \quad \left. \frac{\partial^2 \Omega}{\partial \alpha^2} \right|_0 = \frac{\Omega_2 - 2\Omega_0 + \Omega_4}{(\Delta \alpha)^2}.
\end{align*}
\]

(2.11) (2.12) (2.13)

In (1.9), set

\[
\left. \frac{\partial \phi}{\partial r} \right|_0 = \frac{\phi_1 - \phi_0}{\Delta r}.
\]

(2.14)

Then, in (1.10), use
\begin{equation}
(2.15) \quad \left(1 - \frac{\partial \phi}{\partial \alpha}\right)_0 = \frac{2\Delta \alpha - \phi_2 + \phi_4}{2\Delta \alpha}, \quad \left(\frac{\partial \phi}{\partial r}\right)_0 = \frac{\phi_1 - \phi_3}{2\Delta r}.\end{equation}

Now, define $\gamma$ and $\delta$ by

\begin{equation}
(2.16) \quad \phi_1 - \phi_3 = \gamma, \quad 2\Delta \alpha - \phi_2 + \phi_4 = \delta
\end{equation}

and approximate $\frac{\partial w}{\partial \alpha}$ and $\frac{\partial w}{\partial r}$ in (1.10) as follows:

\begin{equation}
(2.17) \quad \frac{\partial w}{\partial \alpha} = \begin{cases} 
\frac{w_2 - w_0}{\Delta \alpha}, & \gamma \geq 0 \\
\frac{w_0 - w_4}{\Delta \alpha}, & \gamma < 0 
\end{cases}
\end{equation}

\begin{equation}
(2.18) \quad \frac{\partial w}{\partial r} = \begin{cases} 
\frac{w_1 - w_0}{\Delta r}, & \delta \geq 0 \\
\frac{w_0 - w_3}{\Delta r}, & \delta < 0.
\end{cases}
\end{equation}

With respect to (1.11), use (2.13), (2.14), (2.15) and, with $w$ replaced by $\Omega$, (2.17) and (2.18). Finally, in (1.11), approximate $\frac{\partial w}{\partial r}$ and $\frac{\partial w}{\partial \alpha}$ by

\begin{equation}
(2.19) \quad \frac{\partial w}{\partial r} = \frac{w_1 - w_3}{2\Delta r}, \quad \frac{\partial w}{\partial \alpha} = \frac{w_2 - w_4}{2\Delta \alpha}.
\end{equation}

If one defines the quantities $E$, $F$, $G$, $H$, $I$, and $J$ by
\[
E = -\frac{2}{(\Delta r)^2} - \frac{2}{r^2(\Delta \alpha)^2} - \frac{|\gamma|^2}{2 r \Delta r \Delta \alpha} \\
F = \frac{1}{(\Delta r)^2} + \frac{|\delta|^2}{2 r \Delta r \Delta \alpha}, \quad H = \frac{1}{(\Delta r)^2} \\
G = \frac{1}{r^2(\Delta \alpha)^2} + \frac{|\gamma|^2}{2 r \Delta r \Delta \alpha}, \quad I = \frac{1}{r^2(\Delta \alpha)^2} \\
J = w_0 \sin \alpha \left( \frac{w_1 - w_3}{2 \Delta r} \right) + \frac{w_0 \cos \alpha}{r} \left( \frac{w_2 - w_4}{2 \Delta \alpha} \right),
\]
then the respective difference approximations of (1.9) - (1.11) which thereby result are

\[
(2.20) \quad \left[ -\frac{2}{(\Delta r)^2} - \frac{2}{r^2(\Delta \alpha)^2} - \frac{1}{r \Delta r} \right] \phi_0 + \left[ \frac{1}{(\Delta r)^2} + \frac{1}{r \Delta r} \right] \phi_1 + \frac{1}{r^2(\Delta \alpha)^2} \phi_2 \\
+ \frac{1}{(\Delta r)^2} \phi_3 + \frac{1}{r^2(\Delta \alpha)^2} \phi_4 = -\Omega_0,
\]

\[
(2.21) \quad \begin{cases} 
Ew_0 + Fw_1 + Gw_2 + Hw_3 + Iw_4 = -D; & \gamma \geq 0, \delta \geq 0 \\
Ew_0 + Hw_1 + Gw_2 + Fw_3 + Iw_4 = -D; & \gamma \geq 0, \delta < 0 \\
Ew_0 + Fw_1 + Iw_2 + Hw_3 + Gw_4 = -D; & \gamma < 0, \delta \geq 0 \\
Ew_0 + Hw_1 + Iw_2 + Fw_3 + Gw_4 = -D; & \gamma < 0, \delta < 0,
\end{cases}
\]
\[
\begin{align*}
\Omega_0 + F \Omega_1 + G \Omega_2 + H \Omega_3 + I \Omega_4 &= J; & \gamma \geq 0, \ \delta \geq 0 \\
\Omega_0 + H \Omega_1 + G \Omega_2 + F \Omega_3 + I \Omega_4 &= J; & \gamma \geq 0, \ \delta < 0 \\
\Omega_0 + F \Omega_1 + I \Omega_2 + H \Omega_3 + G \Omega_4 &= J; & \gamma < 0, \ \delta \geq 0 \\
\Omega_0 + H \Omega_1 + I \Omega_2 + F \Omega_3 + G \Omega_4 &= J; & \gamma < 0, \ \delta < 0 .
\end{align*}
\]
FIGURE 2
3. The Numerical Method

As shown in Figure 2, let \( R \) be the semicircular region defined by

\[
0 < r < 1, \quad 0 < \alpha < \pi
\]

and let \( S \) be the boundary of \( R \). For finite positive grid sizes \( \Delta r \) and \( \Delta \alpha \), where unity is an integral multiple of \( \Delta r \) and \( \frac{\pi}{2} \) is an integral multiple of \( \Delta \alpha \), construct and number in the usual way the interior polar grid points \( R_h \) and the boundary polar grid points \( S_h \).

In general, we will construct on \( R_h \cup S_h \) a triple sequence of discrete functions

\[
\begin{align*}
(3.1) & \quad \phi^{(0)}, \phi^{(1)}, \phi^{(2)}, \ldots \ldots \\
(3.2) & \quad w^{(0)}, w^{(1)}, w^{(2)}, \ldots \ldots \\
(3.3) & \quad \Omega^{(0)}, \Omega^{(1)}, \Omega^{(2)}, \ldots \ldots ,
\end{align*}
\]

with the property that, for some integral value \( k \), and for given positive tolerances \( \varepsilon_1, \varepsilon_2, \varepsilon_3 \),

\[
\begin{align*}
(3.4) & \quad |\phi^{(k)} - \phi^{(k+1)}| < \varepsilon_1 \\
(3.5) & \quad |w^{(k)} - w^{(k+1)}| < \varepsilon_2 \\
(3.6) & \quad |\Omega^{(k)} - \Omega^{(k+1)}| < \varepsilon_3
\end{align*}
\]
uniformly on $R_h \cup S_h$. Each of the discrete functions in sequences (3.1) - (3.3) will be called an outer iterate. For $j = 1, 2, \ldots$, each $\phi^{(j)}$ will be a solution of (2.20), each $w^{(j)}$ will be a solution of (2.10) or (2.21), and each $\Omega^{(j)}$ will be a solution of (2.22). Numerical convergence to the tolerances (3.4) - (3.6) will yield the discrete approximate solution $\phi^{(k+1)}$, $w^{(k+1)}$, $\Omega^{(k+1)}$ of $\phi$, $w$, $\Omega$, respectively.

Specifically, the algorithm proceeds in the following fashion, with the origin being expressed in rectangular coordinates and with all other points being expressed in polar coordinates.

**Step 1.** Define $\phi^{(0)}$, $w^{(0)}$, and $\Omega^{(0)}$ arbitrarily on $R_h \cup S_h$ except that $\phi^{(0)} = 0$ on $S_h$, $w^{(0)} = 0$ at each point of $S_h$ for which $r = 1$, and $\Omega^{(0)} = 0$ at each point of $S_h$ which is also a point of the X-axis.

**Step 2.** At each point of $S_h$, set

(3.7) \[ \phi = 0. \]

At each point of $R_h$ for which $r = 1 - \Delta r$, set

(3.8) \[ \phi(1 - \Delta r, \alpha) = \frac{\phi(1 - 2\Delta r, \alpha)}{4}. \]

On the remaining points of $R_h$, write down (2.20) with $\Omega_0$ replaced by $\Omega_0^{(k)}$. Solve the linear algebraic system so generated by SOR [4] with over-relaxation factor $r_\phi$ and denote the solution by $\phi^{(k+1)}$. Then, define $\phi^{(k+1)}$ on $R_h \cup S_h$ by the smoothing formula
(3.9) \[ \phi^{(k+1)} = \xi_1 \phi^{(k)} + (1 - \xi_1) \phi^{(k+1)}, \quad 0 \leq \xi_1 \leq 1. \]

**Step 3.** At each point of \( S_h \) for which \( r = 1 \), set \( w = 0 \). At the origin write down (2.10) with each \( \phi \) replaced by the known value \( \phi^{(k+1)}_1 \) given by (3.9), with \( \phi_4 \) replaced by \( -\phi_2 \), and with \( w_4 \) replaced by \( w_2 \). On the remaining points of \( R_h \), write down (2.21) with \( \phi_1 \) replaced by \( \phi^{(k+1)}_1 \). On the remaining points of \( S_h \), write down (2.21) with \( \phi_1 \) replaced by \( \phi^{(k+1)}_1 \), with \( \phi_4 \) replaced by \( -\phi_2 \) and \( w_4 \) replaced by \( w_2 \) between \( O \) and \( P_1 \), and with \( \phi_2 \) replaced by \( -\phi_4 \) and \( w_2 \) replaced by \( w_4 \) between \( O \) and \( P_3 \).

Solve the linear algebraic system generated above by SOR using \( r \) as over-relaxation factor, and denote the solution by \( w^{(k+1)} \).

Then, define \( w^{(k+1)} \) on \( R_h \cup S_h \) by

(3.10) \[ w^{(k+1)} = \xi_2 w^{(k)} + (1 - \xi_2) w^{(k+1)}, \quad 0 \leq \xi_2 \leq 1. \]

**Step 4.** At each point of \( S_h \) for which \( r = 1 \), set

\[ \Omega^{(k+1)}(1, \alpha) = -\frac{2}{(\Delta r)^2} \phi^{(k+1)}(1 - \Delta r, \alpha). \]

Then define \( \Omega^{(k+1)} \) on this point set by

\[ \Omega^{(k+1)} = \xi_3 \Omega^{(k)} + (1 - \xi_3) \Omega^{(k+1)}, \quad 0 \leq \xi_3 \leq 1. \]

**Step 5.** At the points of \( S_h \) not considered in Step 4, which are all on the X-axis, set \( \Omega = 0 \). At each point of \( R_h \), write down (2.22) with \( \phi_1 \) replaced by \( \phi^{(k+1)}_1 \), with \( w_1 \) replaced by \( w^{(k+1)}_1 \), and with
\( \Omega \) at each boundary point for which \( r = 1 \) determined by (3.11). Solve the linear algebraic system so generated by SOR with over-relaxation factor \( r_\Omega \). Denote the solution by \( \bar{\Omega}^{(k+1)} \). Finally, define \( \Omega^{(k+1)} \) on the point set not included in Step 4 by

\[
(3.12) \quad \Omega^{(k+1)} = \xi^4 \Omega^{(k)} + (1 - \xi^4) \bar{\Omega}^{(k+1)}, \quad 0 \leq \xi^4 \leq 1.
\]

**Step 6.** Do Steps 2-5 for \( k = 0, 1, 2, \ldots \). Terminate when (3.4)-(3.6) are satisfied.

For a complete FORTRAN program of the above algorithm, see Schubert [7].

4. **Examples and Results**

A large variety of examples using the method of Section 3 were run on the UNIVAC 1108 at the University of Wisconsin and a selection of convergent ones in which \( D = 10, 100, 250, 500, 1000, 2000 \) and 5000 are summarized in the TABLE. Economic constraints restricted the grid sizes in each case to \( \Delta r = 0.1 \) and \( \Delta \alpha = \pi/18 \), and no case required more than three minutes of computing time. The input values for \( D = 10 \) were \( \phi^{(0)} = w^{(0)} = \Omega^{(0)} = 0 \). The input values for any other \( D \) in the TABLE were the converged results obtained for the previous value of \( D \). The SOR tolerances associated with \( \phi, w, \)
and $\Omega$ were set at $\frac{1}{20} \varepsilon_i$, $i = 1, 2, 3$, respectively. Graphs of constant-$\phi$ and constant-$w$ curves for $D = 10, 100, 500, 2000$ and $5000$ are given in Figures 3-12.

Most of the qualitative physical trends observed by McConalogue and Srivastava continue to develop so that, with increasing $D$, the axial-momentum peak moves well away from the origin, the secondary-flow velocity becomes more uniform in a large central region, and there is a considerable reduction in the flux in the curved tube compared to that of the straight tube. The unexpected result is that the core of the constant-$\phi$ curves exhibits a clockwise motion about the origin up to $D = 500$ and then, for $D \geq 500$, reverses to one which is counterclockwise. It is also of interest to note that, for $D = 5000$, the constant-$w$ curves have developed several oscillatory portions near the origin, which would seem to presage the onset of turbulence.
<table>
<thead>
<tr>
<th>Number Outer Iterations for Convergence</th>
<th>0.3</th>
<th>0.3</th>
<th>0.2</th>
<th>0.2</th>
<th>0.2</th>
<th>0.2</th>
<th>0.1</th>
<th>0.1</th>
<th>0.1</th>
<th>0.1</th>
<th>0.1</th>
<th>0.1</th>
<th>0.1</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergence</td>
<td>0.3</td>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iterations</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Number Outer</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Iteration</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Figure 9 - Constant-w Curves for D = 100
References


Appendix: FORTRAN Program for Secondary Flow, by A. B. Schubert

C TURFLOW  A PROGRAM WHICH COMPUTES THE SECONDARY
C (CROSS-SECTIONAL) FLOW OF A FLUID IN A CURVED TUBE AND THE
C VELOCITY AND FLUX IN THE AXIAL DIRECTION OF THE TUBE BY
C DISCRETIZATION OF THE CROSS-SECTION AND OF THE GOVERNING
C DIFFERENTIAL EQUATIONS.

PARAMETER NR=12, NA=18, NRP1=NR+1, NAP1=NA+1, NRM1=NR-1, NAM1=NA-1,
* NAH=NA/2+1

C NR = NUMBER OF GRID SPACES ON (0,1) IN RADIAL (R) DIRECTION.
C NA = NUMBER OF GRID SPACES ON (0,PI) IN ANGULAR (ALPHA) DIRECTION.

C PRINCIPAL discrete functions computed are defined as follows.
C PHI(I,J,K) = NON-DIMENSIONAL STREAM FUNCTION VALUE FOR SECONDARY
C FLOW AT POLAR GRID POINT CORRESPONDING TO I-TH
C RADIAL VALUE (WHERE I = 1 CORRESPONDS TO R = 0 AND
C 0 = NR+1 CORRESPONDS TO R = 1) AND J-TH ANGULAR
C VALUE (WHERE J = 1 CORRESPONDS TO ALPHA = 0 AND
C J = NA+1 CORRESPONDS TO ALPHA = PI).
C K = 1 IMPLIES A VALUE AT THE PREVIOUS OUTER
C iteration.
C K = 2 IMPLIES A VALUE AT THE PREVIOUS INNER iteration.
C K = 3 IMPLIES A VALUE AT THE CURRENT OUTER AND INNER
C iteration.
C PHI(I,J,K) = NON-DIMENSIONAL VELOCITY IN THE AXIAL DIRECTION,
C WITH SUBSCRIPTS DEFINED AS FOR PHI.
C OMEGA(I,J,K) = INTERMEDIATE VARIABLE INTRODUCED IN ANALYTICAL
C DESCRIPTION OF PROBLEM, WITH SUBSCRIPTS DEFINED
C AS FOR PHI.

DIMENSION PHI(NRP1,NAP1,4), W(NRP1,NAP1,4), OMEGA(NRP1,NAP1,4),
* VI(4), RHOC(3), EPS(3), SA(NAP1), COSA(NAP1), RINV(NRP1), RINV2(NRP1),
* PHIC(3), R(NRP1,5), C(NRP1,NAP1), EPS(3), XIC(4), E(NRP1,NAP1,6),
* EF(NRP1), EF(NRP1), CA(NRP1,NAP1), DELA(NR1), CA(NR1), CD15

DATA PI/3.14159265/, MAXSOR, MAXOUT/250.60/, NOP/3/, (CDR(I), I=1,3)
*/2.2, 2.2, 2.2, ISTART/1/

C MAXSOR = MAXIMUM NUMBER OF ITERATIONS TO BE ALLOWED FOR COMPUTING
C SOLUTION TO ANY SYSTEM BY SUCCESSIVE OVER-RELAXATION.
C MAXOUT = MAXIMUM NUMBER OF OUTER ITERATIONS TO BE ALLOWED IN THE
C COMPUTATION OF PHI, W, AND OMEGA.
C NOP = NUMBER OF DIFFERENT OVER-RELAXATION FACTORS TO BE TRIED
C IN ANY DATA CASE.
C DCR(I) = AMOUNT OF DECREASE IN OVER-RELAXATION FACTOR AT I-TH
C CHANGE.
C ISTART = INPUT CONTROL VARIABLE INDICATING WHETHER OR NOT THE
C FIRST OUTER ITERATES ARE TO BE READ FROM CARDS.
C ISTART.EQ.0 MEANS STARTING VALUES WILL BE SET = 0 BY
C PROGRAM.
C ISTART.NE.0 MEANS STARTING ITERATES WILL BE READ FROM
CARDS IN THE FORMAT (2E10.5).

TEST FOR NECESSITY OF READING STARTING OUTER ITERATES FOR PHI, h, AND OMEGA.
DSTART = VALUE OF PARAMETER D FOR WHICH THE STARTING VALUES ARE A SOLUTION.

IF(ISTART.NE.2) READ(5,31) DSTART,((PHI(I,J,4),J=1,NAP1),I=1,NRP1)
* +((J(I,J,4),J=1,NAP1),I=1,NRP1),((OMEGA(I,J,4),J=1,NAP1),I=1,NRP1)
KRP1,NRP1

DR = GRID SPACING IN RADIAL DIRECTION.
DA = GRID SPACING IN ANGULAR DIRECTION.

DP=1./NR
DA=PI/NA
DH=5.*DA
DRH=5.*DR
DRAH=DR*DA
DRDA=DR/DA
DADR=DA/DR

COMPUTE ELEMENTS OF AREAS DELA(I),I=1,...,NR, IN RADIAL DIRECTION FOR USE IN FLUX CALCULATION.

DO 1 I=1, NR
  1 DELA(I)=(2*I-1)*DRH*DRAH

ON THE GRID DEFINED BY DR AND DA, COMPUTE DISCRETE FUNCTIONS DEPENDENT ONLY ON RADIUS AND ANGLE.

SA(1)=C.
DO 2 J=2,NA
  2 SA(J)=SIN((J-1)*DA)*DA
2 CGSA(J)=COS((J-1)*DA)
SA(NAP1)=0.
COSA(NAP1)=-1.
RINV(NRP1)=1.
RINV2(NRP1)=1.
DO 3 I=2, NR
  3 RINV(I)=1./((I-1)*DR)
DRRH=DRH*RINV(I)
RINV2(I)=RINV(I)**2
DO 3 J=2, NA
3 CGAT+J=DRRH*COSA(J)

COMPUTE SCR COEFFICIENTS FOR PHI.

DO 4 I=2, NR
  4 BC=2.*(DA+RDA+RINV2(I))*DA*RINV(I)
B(I,1)=(DA+RDA+RINV(I))/BC
B(I,2)=RDA*RINV2(I)/BC
B(I,3)=DA/RD
B(I,4)=B(I,2)
READ OUTER ITERATION SMOOTHING PARAMETERS (XI), OVER-RELAXATION FACTORS (RHOC), OUTER ITERATION CONVERGENCE TOLERANCES (EPPS), PARAMETER RELATED TO REYNOLDS NO. (D), AND INPUT/OUTPUT CONTROL VARIABLES DEFINED AS FOLLOWS.

ISAVE=NEQ implies save the solution for this case (if obtained) in memory for use as starting values for a succeeding case with a different value of D.

ISAVE=FB=NEQ implies do not do the above.

ISAVE=FUQ implies take starting values for outer iteration from memory.

IPCH=NEQ implies initialize outer iterates to zero.

IPCH=FBQ implies punch solution for this case (if obtained) out on cards in the format (3E10.5).

IPCH=FBQ implies do not punch solution obtained for this case.

READ(5,6,END=70) XI,RHOC,EPPS,D,ISAVE,IPCH

PRINT OUT INPUT PARAMETERS.

WRITE(5,99) D,RHOC,XI,EPPS

COMPUTE PARAMETERS DEPENDENT ON THE INPUT PARAMETER D, INCLUDING THE COEFFICIENTS, CO(I), I=1,...,NR, IN THE NUMERICAL INTEGRATION FOR THE FLUX.

CORDAM=D*CORDAM
DOR2=D.OR**2
CON=4.*PI*D
CO(1)=1E6*DELA(1)/(2.*PI**2)
DO 105 I=2,NR
105 CO(I)=CON+DELA(I)

COMPUTE COMPLEMENTS (RELATIVE TO 1) OF SMOOTHING PARAMETERS AND OVER-RELAXATION FACTORS.

ALSO COMPUTE FOR CONVERGENCE TOLERANCES, EPPS(I), I=1,2,3, AS FUNCTIONS OF OUTER ITERATION TOLERANCES, EPS(I), I=1,2,3.

DO 2 I=1,3
XI0(I)=1.-XI(I)
RHOC0(I)=1.-RHOC(I)
5 EPPS0(I)=.05*EPS0(I)
XI0(4)=1.-XI(4)

TEST WHETHER INITIAL OUTER ITERATES ARE TO BE OBTAINED FROM MEMORY, HAVING BEEN PLACED THERE AS THE SOLUTION FOR A PREVIOUS VALUE OF D EITHER BY CARD INPUT OR BY A PREVIOUS COMPUTATION THIS RUN.

IF(IPCH.EQ.0) GO TO 7
DO 106 I=1,NRP1
DO 106 J=1,NRP1
PHI(I,J,3)=PHI(I,J,4)

106 CONTINUE

7 CONTINUE
C INTEGER OUTER ITERATES TO ZERO IF NEITHER INPUT NOR COMPUTED
C PREVIOUSLY THIS RUN.

7 DO 9 I=1,NR11
   DO 9 J=1,NR11
      PHI(I,J,3)=C.
      W(I,J,3)=C.
9 CONTINUE

C INITIALIZE OUTER ITERATION COUNTER (ICV) AND COUNTERS OF NUMBER
C OF OVER-RELAXATION FACTORS USED FOR PHI AND OMEGA (IR,IRG, RESP.)
C FOR THIS DATA CASE.

C IOUT=0
C IR=0
C IRG=0

C TEST FOR MAXIMUM NUMBER OF OUTER ITERATIONS.

10 IF(IOUT .GE. MAXOUT) GO TO 58

C UPDATE OUTER ITERATION COUNTER, WRITE MESSAGE INDICATING NEW
C OUTER ITERATION NUMBER, AND SET OUTER ITERATION CONVERGENCE
C INDICATOR (ICV) TO ZERO. AFTER COMPUTATION OF ALL CURRENT OUTER
C ITERATES AND COMPARISON WITH PREVIOUS OUTER ITERATES AGAINST THE
C SPECIFIED TOLERANCES, ICV WILL STILL BE ZERO IF CONVERGENCE HAS
C BEEN OBTAINED, OTHERWISE ICV WILL WILL BE NON-ZERO.

IOUT=IOUT+1
WRITE(C,82) IOUT
ICV=0

C UPDATE PREVIOUS OUTER ITERATES.

DO 12 I=1,NR11
   DO 12 J=1,NR11
      PHI(I,J,1)=PHI(I,J,3)
      W(I,J,1)=W(I,J,3)
12 CONTINUE

C COMPUTE CONSTANT SOR COEFFICIENT FOR PHI.

DO 13 I=1,NI
   DO 13 J=1,NA
      C(I,J)=C(I,J)*OMEG(A(I,J,1))
13 CONTINUE

C INITIALIZE SOR ITERATION COUNTER FOR PHI.
ISOR=0
C TEST FOR MAXIMUM NUMBER OF SOR ITERATIONS FOR PHI.

10 IF(ISOR.GT.MAXSOR) GO TO 55

C UPDATE SOR ITERATION COUNTER IF MAXIMUM NUMBER HAS NOT BEEN
C ACHIEVED.

ISOR=ISOR+1

C UPDATE PREVIOUS SOR ITERATES FOR PHI.

DO 15 I=2, NR
  DO 10 J=2, NA
15 PHI(I,J,2)=PHI(I,J,3)

C SET SOR ITERATION CONVERGENCE INDICATOR (ICONV) TO ZERO. AFTER
C COMPUTATION OF CURRENT SOR ITERATE AND COMPARISON WITH PREVIOUS
C ITERATE AGAINST THE TOLERANCE, ICONV WILL STILL BE ZERO IF
C CONVERGENCE HAS BEEN OBTAINED, OTHERWISE IT WILL BE NON-ZERO.

ICONV=C

C COMPUTE CURRENT SOR ITERATE FOR PHI...
C ...FIRST ON INTERIOR OF REGION, EXCLUDING 'INNER BOUNDARY'
C R = 1.-DR.

DO 18 I=2, NR
  DO 13 J=2, NA
    PHI(I,J,3)=RHOC(1)*PHI(I,J,2)+RHO(1)*(E(I,1)*PHI(I+1,J,2)
     +G(I,1)*PHI(I,J+1,2)+S(I,2)*PHI(I-1,J,2)+B(I,4)*PHI(I,J-1,3)
     +C(I,J))
    IF(AES(PHI(I,J,2)-PHI(I,J,3)).GT.EPS(1)) ICONV=1
  13 CONTINUE

C ...THEN ON 'INNER BOUNDARY' R = 1.-DR.

DO 19 J=2, NA
  PHI(NR,J,3)=RHOC(1)*PHI(NR,J,2)+RHO(1)*.25*PHI(NRM1,J,3)
  IF(AES(PHI(NR,J,2)-PHI(NR,J,3)).GT.EPS(1)) ICONV=1
  19 CONTINUE

C TEST FOR CONVERGENCE OF SOR ITERATION FOR PHI.

IF(ICONV.NE.3) GO TO 14

C SOR CONVERGENCE ATTAINED FOR PHI, SMOOTH OUTER ITERATE.

DO 20 I=2, NR
  DO 10 J=2, NA
    PHI(I,J,3)=XI(1)*PHI(I,J,1)+XIC(1)*PHI(I,J,3)
    IF(AES(PHI(I,J,1)-PHI(I,J,3)).GT.EPS(1)) TCV=1
  20 CONTINUE
C PRINT OUT SMOOTHED CURRENT OUTER ITERATE FOR PHI.

CALL OUTPUT('PHI',ISOR,PHI(I+1,J))

C ...FIRST COMPUTE COEFFICIENTS AT THE ORIGIN

EQ=4.*ABS(PHI(Z,NAM,2))
EI=1.-AMIN1(PHI(Z,NAM,3),0.)/EC
E2=2./E3
E3=(1.+AMAX1(PHI(Z,NAM,3),0.))/EC
E4=30./E3

C ...NEXT COMPUTE COEFFICIENTS FOR REMAINDER OF REGION...

FIRST FOR RAYS ALONG ALPHA = C AND ALPHA = PI, EXCLUDING
R = C AND R = 1.*

DO 23 I=2,NR
DELTA1=CA-PHI(I,2,3)
DELTA2=DA-PHI(I,NA,3)
EE(I)=2.*(DAE+CAE*RINV2(I))
EE(I)=EE(I)+RINV1(I)*ABS(DELTA1)
EE(I)=EE(I)+RINV2(I)*ABS(DELTA2)
E(I,NAP1,1)=(DAE+RINV1(I)*AMAX1(DELTA1,C,.) )/EE1
E(I,NAP1,1)=(DAE+RINV1(I)*AMAX1(DELTA2,C,.) )/EE2
EE(I)=DRAE*RINV2(I)
E(I,1,2)=EE(I)/EE1
E(I,NAP1,2)=EE(I)/EE2
E(I,NAP1,3)=CAE-RINV1(I)*AMIN1(DELTA1,C,.) )/EE1
E(I,NAP1,3)=CAE-RINV1(I)*AMIN1(DELTA2,C,.) )/EE2
E(I,1,4)=E(I,1,5)
E(I,NAP1,4)=E(I,NAP1,2)
E(I,1,5)=DORDAM/EE1
23 E(I,NAP1,5)=DORDAM/EE2

C ...THEN ON INTERIOR OF REGION.

DO 25 J=2,NA
DO 25 J=2,NA
GAMMA=5.*(PHI(I+1,J,3)-PHI(I-1,J,3))
DELTA=5.* (PHI(I,J+1,3)-PHI(I,J-1,3))
EE(I,J)=EE(I)+RINV1(I)*ABS(GAMMA)*ABS(DELTA))
E(I,J,1)=(DAE+RINV1(I)*AMAX1(DELTA,C,.) )/E(I,J,6)
E(I,J,2)=(EE(I)+RINV2(I)*AMAX1(GAMMA,C,.) )/E(I,J,5)
E(I,J,3)=(DAE-RINV1(I)*AMIN1(DELTA,C,.) )/E(I,J,6)
E(I,J,4)=(EE(I)-RINV2(I)*AMIN1(GAMMA,C,.) )/E(I,J,5)
25 E(I,J,5)=DORDAM/E(I,J,6)

C INITIALIZE SGR ITERATION COUNTER FOR J.

26 ISOR=0

C TEST FOR MAXIMUM NUMBER OF SGR ITERATIONS.
27 IF(ISOR.GE.MAXSOR) GO TO 56
C UPDATE SOR ITERATION COUNTER IF MAXIMUM HAS NOT BEEN ACHIEVED.
ISOR=ISOR+1
C UPDATE PREVIOUS SOR ITERATES FOR W...
C ...FIRST AT THE ORIGIN
W(I+1,J)=W(I+1,J)
C ...THEN ON REMAINDER OF REGION, EXCLUDING R = 1.
DO 29 I=2,NI
DO 29 J=1,NAJ
29 W(I+J,J)=W(I+J,J)
C SET SOR CONVERGENCE INDICATOR (ICONV) TO ZERO.
ICONV=0
C COMPUTE CURRENT SOR ITERATE FOR W...
C ...FIRST AT THE ORIGIN
W(I+1,J)=RHO(I)*W(I+1,J) + RHO(2)*W(I+1,J)*E1*W(I+1,J)+E2*W(I+1,J)+E3*W(I+1,J)+E4
C (SET VALUES OF W FOR R = 0 AND ALL ANGLES ALPHA(I), I=1,...,NA1, EQUAL TO VALUE OF W AT ORIGIN FOR CONVENIENCE IN SOR COMPUTATION.)
DO 30 J=2,NAJ
30 W(I+J,J)=W(I+1,J)
IF(ABS(W(I+1,J)-W(I+1,J)).GT.EPS) ICONV=1
C ...NEXT ALONG RAY ALPHA = 0, EXCLUDING R = 0 AND R = 1.
DO 32 I=2,NI
32 W(I+1,J)=RHO(I)*W(I+1,J) + RHO(2)*W(I+1,J)*E(I)*W(I+1,J)+E(I)*W(I+1,J)+E(I)*W(I+1,J)+E(I)*W(I+1,J)
IF(ABS(W(I+1,J)-W(I+1,J)).GT.EPS) ICONV=1
32 CONTINUE
C ...NEXT ON INTERIOR OF REGION
DO 33 I=2,NI
DO 33 J=2,NA
IF(ABS(W(I+J,J)-W(I+J,J)).GT.EPS) ICONV=1
33 CONTINUE
C ...FINALLY ALONG RAY ALPHA = PI, EXCLUDING R = 0 AND R = 1.
DO 35 I=2,NI
\( u(I, NA1, 3) = RHOC(2) \times u(I, NA1, 2) + RHOC(2) \times (E(I, NA1, 1) \times u(I, NA1, 2) + E(I, NA1, 3) \times u(I-1, NA1, 2) + E(I, NA1, 4) \times u(I, NA1, 3)) + E(I, NA1, 5) \times u(I, NA1, 3) \)

IF (ABS(u(I, NA1, 2) - u(I, NA1, 3)) \times GT.EPS(2)) ICONV = 1

34 CONTINUE

C TEST FOR CONVERGENCE OF SOR ITERATION FOR U.

IF (ICONV.NE.3) GO TO 27

C SOR CONVERGENCE ACHIEVED FOR U. SMOOTH OUTER ITERATE...

C ... FIRST AT THE ORIGIN

\( u(1, 1, 3) = X(I, 1, 2) \times u(1, 1, 1) + XIC(2) \times u(1, 1, 3) \)

C (SET VALUES OF U FOR R = 2 AND ALL DISCRETE ANGLES EQUAL TO

C SMOOTHED VALUE OF U AT ORIGIN.)

DO 134 J = 1, NA1

134 \( u(I, J, 3) = u(I, J, 1) \)

IF (ABS(u(I, J, 1) - u(I, J, 3)) \times GT.EPS(2)) ICV = 1

C ... THEN ON REMAINDER OF REGION, EXCLUDING R = 1.

DO 33 I = 2, NR

DO 35 J = 1, NA1

\( u(I, J, 3) = X(I, 1, 2) \times u(I, J, 1) + XIC(2) \times u(I, J, 3) \)

IF (ABS(u(I, J, 1) - u(I, J, 3)) \times GT.EPS(2)) ICV = 1

35 CONTINUE

C PRINT OUT SMOOTHED CURRENT OUTER ITERATE FOR U.

CALL OUTPUT('U', ISOR, u(I, J, 3))

C COMPUTE AND SMOOTH OMEGA ON BOUNDARY R = 1.

DO 37 J = 1, NA1

\( \omega(2 + J, 1) = X(I, 1, 2) \times \omega(1 + J, 1) \times XIC(2) \times \omega(1, J, 3) \)

IF (ABS(\( \omega(1 + J, 1) - \omega(1, J, 3) \)) \times GT.EPS(2)) ICV = 1

37 CONTINUE

C COMPUTE CONSTANT COEFFICIENT FOR SOR ITERATION FOR OMEGA.

DO 40 I = 2, NR

DO 48 J = 1, NA1

\( C(I, J, 3) = (X(I, J, 1) \times (5A(J) \times (u(I, J, 3) - u(I-1, J, 3)) + CA(I, J)) + (u(I, J + 1, 3) - 4(u(I, J-1, 3))) / E(I, J, 5) \)

C INITIALIZE COUNTER OF SOR ITERATION NUMBER FOR OMEGA.

41 ISOR = 0

C TEST FOR MAXIMUM NUMBER OF SOR ITERATIONS.
DO 44 I=2, NR
DO 44 J=2, NA
44 OMEGA(I,J,2)=OMEGA(I,J,3)

C SET SOR CONVERGENCE INDICATOR TO ZERO.
ICONV=0

C COMPUTE CURRENT SOR ITERATE FOR OMEGA ON INTERIOR.
DO 46 I=2, NR
DO 46 J=2, NA
OMEGA(I,J,3)=RHOC(7)*OMEGA(I,J,2)+
  (E(I,J,1)+E(I,J,4))*OMEGA(I,J,2)*
  E(I,J,2)*OMEGA(I,J,1)+E(I,J,3)*OMEGA(I,J,2)+
  E(I,J,4)*OMEGA(I,J,2)+
  *OMEGA(I,J-1,3)+E(I,J,6)
IF(AES(OMEGA(I,J,2)-OMEGA(I,J,3)).GT.EEPS(3)) ICONV=1
46 CONTINUE

C TEST FOR CONVERGENCE OF SOR ITERATION FOR OMEGA.
IF(ICONV.NE.0) GO TO 42

C SOR CONVERGENCE ATTAINED FOR OMEGA, SMOOTH OUTER ITERATE.
DO 43 I=2, NR
DO 43 J=2, NA
OMEGA(I,J,2)=XIC(4)*OMEGA(I,J,1)+XIC(4)*OMEGA(I,J,3)
IF(AES(OMEGA(I,J,1)-OMEGA(I,J,3)).GT.EEPS(3)) ICV=1
43 CONTINUE

C PRINT OUT SMOOTHED CURRENT OUTER ITERATE FOR OMEGA.
CALL OUTPUT('OMEGA',ISOR,OMEGA(I,J,3))

C TEST FOR CONVERGENCE OF ALL OUTER ITERATES.
IF(ICV.NE.0) GO TO 10

C PRINT MESSAGE INDICATING CONVERGENCE OF OUTER ITERATION.
WRITE(*,377)
377 C PUNCH SOLUTION (AND VALUE OF D) OUT ON CARDS IF INPUT CONTROL
C PARAMETER IPECH 50 DICTIONS, AND PRINT MESSAGE INDICATING THAT
C THIS PUNCHING WAS DONE.
IF(IFCH.NF.GT.0) PRINT(300) ((PHI(I,J,J3)=J=1,NAP1),I=1,NRP1),((N(I,J,J3)*J=1,NAP1),I=1,NRP1)
IF(IFCH.NF.GT.0) WRITE(6,86)

C COMPUTE RATIO OF FLUX IN CURVED TUBE TO THAT IN STRAIGHT TUBE.

QR=0.
DO 40 J=1,NA
40 QR=QR+K(1,J,3)+N(2,J,3)+N(2,J+1,3)
QR=QR*CO(1)
DO 240 I=1,MR
S=0.
DO 140 J=1,NA
140 S=S+PO(I,J,3)+N(I,J,3)+W(I,J+1,3)+W(I+1,J+1,3)
240 GR=GR+CO(I)*5

C PRINT OUT VALUE OF FLUX RATIO JUST COMPUTED.

WRITE(6,99) QR

C TEST WHETHER OR NOT SOLUTION JUST OBTAINED IS TO BE SAVED IN
C MEMORY FOR SOME SUCCEEDING CASE WITH A DIFFERENT VALUE OF D.

IF(ISAVE.EQ.0) GO TO 5

C SAVE CURRENT SOLUTION IN ANOTHER AREA OF MEMORY.

DO 50 I=1,NRP1
DO 50 J=1,NAP1
PHI(I,J,4)=PHI(I,J,3)
N(I,J,4)=N(I,J,3)
50 OMEGAI(J,4)=OMEGAI(J,3)

C SAVE VALUE OF D FOR WHICH SOLUTION WAS JUST OBTAINED IN OSTART.

OSTART=0
GO TO 5

C CONVERGENCE FAILURE MESSAGES

C PHI ITERATION FAILED. READ NEXT DATA CASE.

55 WRITE(6,100)
GO TO 5

C S iteration failed. REDUCE OVER-RELAXATION FACTOR AND TRY AGAIN.
C UNLESS THIS HAS ALREADY BEEN DONE THE MAXIMUM NUMBER OF TIMES
C (NOR), IN WHICH CASE READ NEXT DATA CASE.

56 WRITE(6,100) RH0(2)
IF(IRU.GE.NOR) GO TO 5
IRU=IRU+1
RH0(2)=RH0(2)-DOR(IRU)
RHOC(2)=1.-RHOC(2)
GO TO 155
155 I=1,NAP1
GO TO 155
156 M(I,J)=M(I,J,1)
GO TO 20

C OMEGA ITERATION FAILED. REDUCE D-R FACTOR AND TRY AGAIN, UNLESS
C THIS HAS ALREADY BEEN DONE THE MAXIMUM NUMBER OF TIMES (NDR),
C IN WHICH CASE READ THE NEXT DATA CASE.

57 WRITE(6,24) RH0(3)
IF(IRD.GE.NDR) GO TO 5
IRD=IRD+1
RH0(3)=RHO(3)-COR(IRD)
RHO(3)=1.-RH0(3)
GO TO 157
157 OMEGA(I,J)=OMEGA(I,J,1)
GO TO 51

C OUTER ITERATION FAILED. PRINT MESSAGE INDICATING THIS AND READ
C NEXT DATA CASE.

58 WRITE(6,30)
GO TO 5

C TERMINATION POINT FOR PROGRAM. CONTROL REACHES HERE AFTER ATTEMPT
C TO READ PAST LAST DATA RECORD.

70 STOP

59 FORMAT(11F5.5,3(1X))
33 FORMAT(1X,'I0 = ',F5.3,' I1 = ',F5.3,' I2 = ',F5.3,' I3 = ',F5.3,' I4 = ',F5.3)
25 FORMAT('SOR FOR PHI FAILED. ')
25 FORMAT('SOR FOR PHI FAILED WITH SOR FACTOR = ',F5.2)
24 FORMAT('SOR FOR OMEGA FAILED WITH SOR FACTOR = ',F5.2)
22 FORMAT('OUTER ITERATION FAILED TO CONVERGE.')
2 I FORMAT('OUTER ITERATION IS// ')
2 I FORMAT(1X,11F11.8)
23 FORMAT('SOR FOR PHI FACTOR FOR I CHANGED TO ',F5.2)
23 FORMAT('SOR FOR PHI FACTOR FOR I CHANGED TO ',F5.2)
27 FORMAT('INITIAL ITERATE TAKEN FROM SOLUTION FOR DI = ',F7.2)
25 FORMAT('OUTER ITERATION HAS OUTPUT ON PUNCHED CARDS.')

END

C THIS ROUTINE PRINTS OUT A DISCRETE FUNCTION IN A RECTANGULAR
FORMAT WHICH IS RELATED TO THE FOLLOWING WAY TO THE POLAR GRID
IMPOSED ON THE PHYSICAL REGION.

THE TOP LINE OF THE PRINT BLOCK CORRESPONDS TO VALUES OF THE
FUNCTION ALONG THE ARC $R = 1$.

THE BOTTOM LINE OF THE PRINT BLOCK CORRESPONDS TO THE VALUE
OF THE FUNCTION AT THE ORIGIN ($R = 0$).

SUBROUTINE OUTPUT(VAR, ISOR, A)
PARAMETER NR=10, NA=13, NRF1=NRF+1, NAF1=NA+1
DIMENSION A(NRF1, NAF1)
DATA PHI, A, AOMEGA,'PHI ', 'W ', 'OMEGA '/
30 FORMAT(1HC, 5X 'ISOR ITERATIONS'/)
90 FORMAT(1X F6.2, 17F7.2, F6.2)
97 FORMAT(1X F6.1, 17F7.1, F6.1)

KRF1=NRF1
KAPI=NAF1
WRITE(6, 90) VAR, ISOR
IF(VAR, 'W', AOMEGA) GO TO 2
DO 5 I=1, NRF1
5 WRITE(6, 97) (A(KRP1-I+1, KAPI-J+1), J=1, KAPI)
RETURN
3 DO 10 I=1, NRF1
10 WRITE(6, 97) (A(KRP1-I+1, KAPI-J+1), J=1, KAPI)
RETURN
END