WIS-CS-72-155
Computer Sciences Department
UNIVERSITY OF WISCONSIN
1210 West Dayton Street
Madison, Wisconsin 53706

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.

SECONDARY FLOW IN A CURVED TUBE

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 θ with a fixed axial plane. Let OC be of length L, and let the radius of the cross section be $\underline{a}$. The coordinates of any point P of the cross section are denoted by orthogonal coordinates (r', α, θ), where r' is the distance CP and α is the angle CP makes with OC. Let the velocity components at

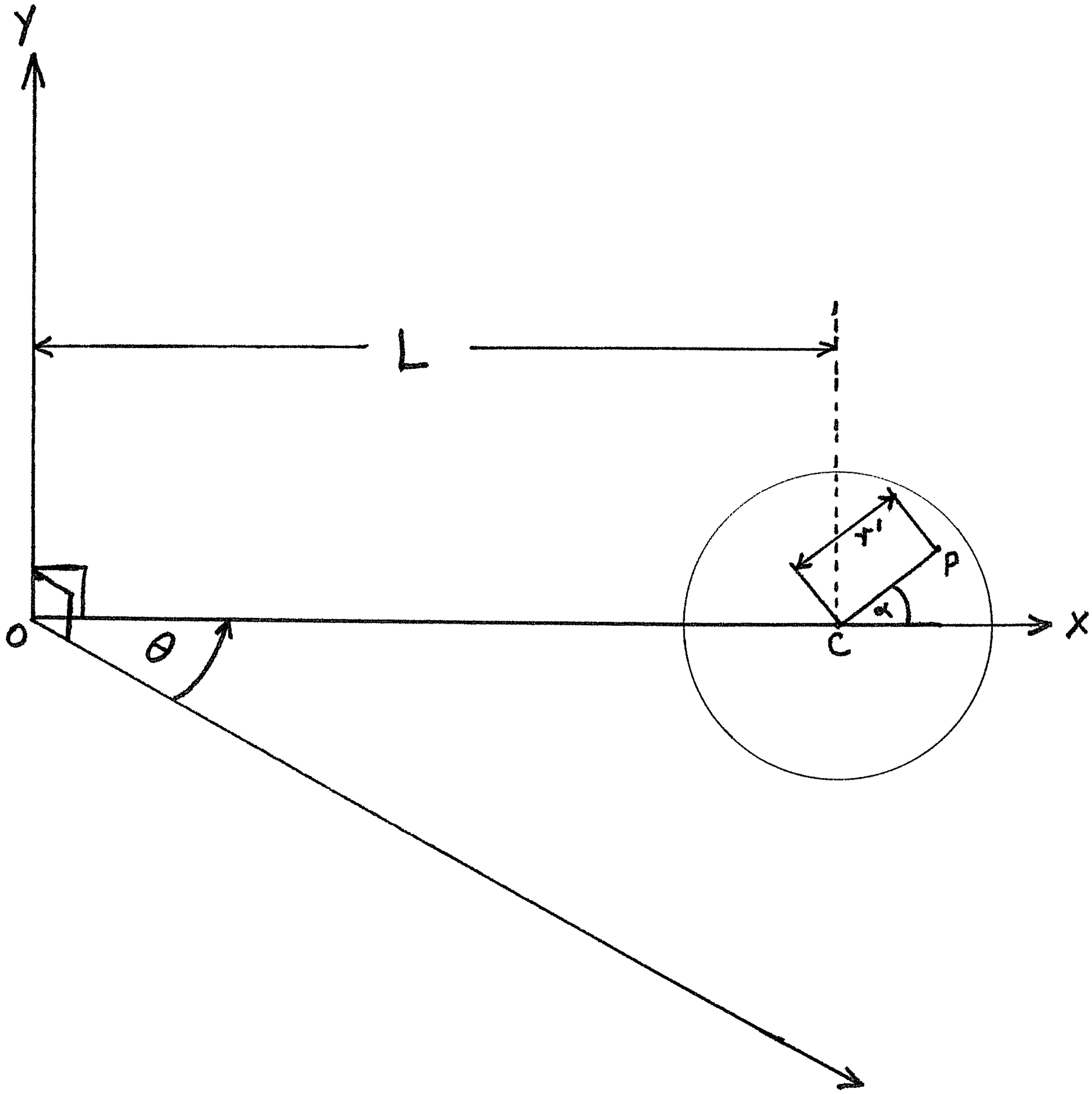

Figure 1

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

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

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

$$(1.2)\qquad D = 4R\sqrt{\frac{2a}{L}},$$

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

$$(1.3)\qquad f = \nu\phi, \quad W = w\left(\frac{\nu^2 L}{2a^3}\right)^{1/2}, \quad r' = ar,$$

where ν is the kinematic viscosity, yields the following equations of motion [6]:

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

$$(1.5)\qquad -\nabla_1^4 \phi = \frac{1}{r}\left(\frac{\partial \phi}{\partial r}\frac{\partial}{\partial \alpha} - \frac{\partial \phi}{\partial \alpha}\frac{\partial}{\partial r}\right)\nabla_1^2 \phi + w\left(\frac{\partial w}{\partial r}\sin\alpha + \frac{\partial w}{\partial \alpha}\frac{\cos\alpha}{r}\right),$$

in which

$$\nabla_1^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \alpha^2} \quad . \tag{1.6}$$

The boundary constraints at $r = 1$ are

$$w = \phi = \frac{\partial \phi}{\partial r} = 0. \tag{1.7}$$

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:

$$0 \leq D \leq 5000. \tag{1.8}$$

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:

$$(1.9) \qquad \frac{\partial^2 \phi}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 \phi}{\partial \alpha^2} + \frac{1}{r}\frac{\partial \phi}{\partial r} = -\Omega$$

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

$$(1.11) \qquad \frac{\partial^2 \Omega}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 \Omega}{\partial \alpha^2} + \frac{1}{r}\left[\frac{\partial \phi}{\partial r}\frac{\partial \Omega}{\partial \alpha} + \left(1 - \frac{\partial \phi}{\partial \alpha}\right)\frac{\partial \Omega}{\partial r}\right]$$

$$= w\left(\sin\alpha \frac{\partial w}{\partial r} + \frac{\cos\alpha}{r}\frac{\partial w}{\partial \alpha}\right) \quad .$$

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

$$(1.9') \qquad \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -\Omega$$

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

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

into polar coordinates.

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

$$(1.12) \qquad \phi(x,y) = -\phi(x,-y)$$

$$(1.13) \qquad \Omega(x,y) = -\Omega(x,-y)$$

$$(1.14) \qquad w(x,y) = w(x,-y),$$

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 rectagular coordinates, that

$$\phi(x,0) = \Omega(x,0) = 0 \, . \tag{1.15}$$

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

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

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) $$\varepsilon = \phi_1 - \phi_3$$

(2.4) $$\beta = \phi_2 - \phi_4\ .$$

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

(2.5) $$\frac{\partial w}{\partial y} = \begin{cases} \dfrac{w_2 - w_0}{\Delta r}\ , & \varepsilon \geq 0 \\ \dfrac{w_0 - w_4}{\Delta r}\ , & \varepsilon < 0 \end{cases}$$

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

If one now defines the quantities A, B, C by

(2.7) $$A = -4 - \frac{|\varepsilon|}{2} - \frac{|\beta|}{2}$$

(2.8) $$B = 1 + \frac{|\varepsilon|}{2}$$

(2.9) $$C = 1 + \frac{|\beta|}{2}\ ,$$

then the difference approximation of (1.10') which results is

$$
(2.10)\quad \begin{cases}
A w_0 + w_1 + B w_2 + C w_3 + w_4 = -(\Delta r)^2 D; & \varepsilon \geq 0, \quad \beta \geq 0 \\
A w_0 + C w_1 + B w_2 + w_3 + w_4 = -(\Delta r)^2 D; & \varepsilon \geq 0, \quad \beta < 0 \\
A w_0 + w_1 + w_2 + C w_3 + B w_4 = -(\Delta r)^2 D; & \varepsilon < 0, \quad \beta \geq 0 \\
A w_0 + C w_1 + w_2 + w_3 + B w_4 = -(\Delta r)^2 D; & \varepsilon < 0, \quad \beta < 0.
\end{cases}
$$

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

$$
(2.11)\qquad \left.\frac{\partial^2\phi}{\partial r^2}\right|_0 = \frac{\phi_1 - 2\phi_0 + \phi_3}{(\Delta r)^2}\,, \qquad \left.\frac{\partial^2\phi}{\partial \alpha^2}\right|_0 = \frac{\phi_2 - 2\phi_0 + \phi_4}{(\Delta\alpha)^2}
$$

$$
(2.12)\qquad \left.\frac{\partial^2 w}{\partial r^2}\right|_0 = \frac{w_1 - 2w_0 + w_3}{(\Delta r)^2}\,, \qquad \left.\frac{\partial^2 w}{\partial \alpha^2}\right|_0 = \frac{w_2 - 2w_0 + w_4}{(\Delta\alpha)^2}
$$

$$
(2.13)\qquad \left.\frac{\partial^2\Omega}{\partial r^2}\right|_0 = \frac{\Omega_1 - 2\Omega_0 + \Omega_3}{(\Delta r)^2}\,, \qquad \left.\frac{\partial^2\Omega}{\partial \alpha^2}\right|_0 = \frac{\Omega_2 - 2\Omega_0 + \Omega_4}{(\Delta\alpha)^2}\,.
$$

In (1.9), set

$$
(2.14)\qquad \left.\frac{\partial\phi}{\partial r}\right|_0 = \frac{\phi_1 - \phi_0}{\Delta r}\,.
$$

Then, in (1.10), use

$$(2.15)\qquad \left(1-\frac{\partial\phi}{\partial\alpha}\right)\bigg|_0 = \frac{2\Delta\alpha - \phi_2 + \phi_4}{2\Delta\alpha}, \quad \left(\frac{\partial\phi}{\partial r}\right)\bigg|_0 = \frac{\phi_1 - \phi_3}{2\Delta r} \quad .$$

Now, define γ and δ by

$$(2.16)\qquad \phi_1 - \phi_3 = \gamma, \quad 2\Delta\alpha - \phi_2 + \phi_4 = \delta$$

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

$$(2.17)\qquad \frac{\partial w}{\partial \alpha} = \begin{cases} \dfrac{w_2 - w_0}{\Delta\alpha}, & \gamma \geq 0 \\[2ex] \dfrac{w_0 - w_4}{\Delta\alpha}, & \gamma < 0 \end{cases}$$

$$(2.18)\qquad \frac{\partial w}{\partial r} = \begin{cases} \dfrac{w_1 - w_0}{\Delta r}, & \delta \geq 0 \\[2ex] \dfrac{w_0 - w_3}{\Delta r}, & \delta < 0. \end{cases}$$

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

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

If one defines the quantities E, F, G, H, I, and J by

$$E = -\frac{2}{(\Delta r)^2} - \frac{2}{r^2(\Delta\alpha)^2} - \frac{|\gamma| + |\delta|}{2r\Delta r\Delta\alpha}$$

$$F = \frac{1}{(\Delta r)^2} + \frac{|\delta|}{2r\Delta r\Delta\alpha} , \quad H = \frac{1}{(\Delta r)^2}$$

$$G = \frac{1}{r^2(\Delta\alpha)^2} + \frac{|\gamma|}{2r\Delta r\Delta\alpha} , \quad I = \frac{1}{r^2(\Delta\alpha)^2}$$

$$J = w_0 \sin\alpha \left(\frac{w_1 - w_3}{2\Delta r}\right) + \frac{w_0 \cos\alpha}{r} \left(\frac{w_2 - w_4}{2\Delta\alpha}\right) ,$$

then the respective difference approximations of (1.9) - (1.11) which thereby result are

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

(2.21) $$\begin{cases} Ew_0 + Fw_1 + Gw_2 + Hw_3 + Iw_4 = -D; & \gamma \geq 0,\ \delta \geq 0 \\ Ew_0 + Hw_1 + Gw_2 + Fw_3 + Iw_4 = -D; & \gamma \geq 0,\ \delta < 0 \\ Ew_0 + Fw_1 + Iw_2 + Hw_3 + Gw_4 = -D; & \gamma < 0,\ \delta \geq 0 \\ Ew_0 + Hw_1 + Iw_2 + Fw_3 + Gw_4 = -D; & \gamma < 0,\ \delta < 0 , \end{cases}$$

$$
(2.22)\quad \begin{cases}
E\Omega_0 + F\Omega_1 + G\Omega_2 + H\Omega_3 + I\Omega_4 = J; & \gamma \geq 0, \ \ \delta \geq 0 \\
E\Omega_0 + H\Omega_1 + G\Omega_2 + F\Omega_3 + I\Omega_4 = J; & \gamma \geq 0, \ \ \delta < 0 \\
E\Omega_0 + F\Omega_1 + I\Omega_2 + H\Omega_3 + G\Omega_4 = J; & \gamma < 0, \ \ \delta \geq 0 \\
E\Omega_0 + H\Omega_1 + I\Omega_2 + F\Omega_3 + G\Omega_4 = J; & \gamma < 0, \ \ \delta < 0 \ .
\end{cases}
$$

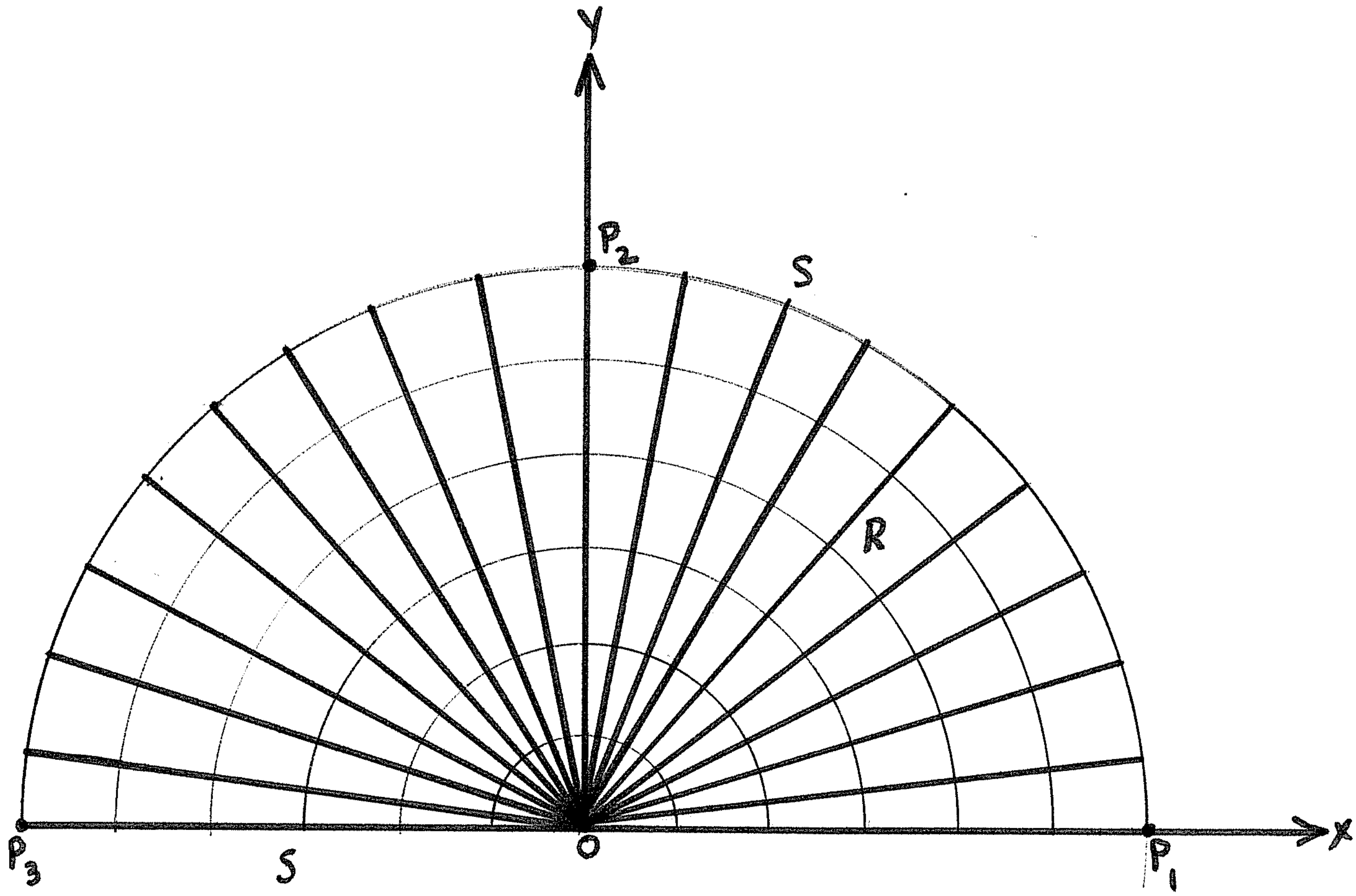

FIGURE 2

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 Δr and $\Delta\alpha$, where unity is an integral multiple of Δr 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

$$(3.1) \qquad \phi^{(0)}, \phi^{(1)}, \phi^{(2)}, \ldots.$$

$$(3.2) \qquad w^{(0)}, w^{(1)}, w^{(2)}, \ldots.$$

$$(3.3) \qquad \Omega^{(0)}, \Omega^{(1)}, \Omega^{(2)}, \ldots. \ ,$$

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

$$(3.4) \qquad |\phi^{(k)} - \phi^{(k+1)}| < \varepsilon_1$$

$$(3.5) \qquad |w^{(k)} - w^{(k+1)}| < \varepsilon_2$$

$$(3.6) \qquad |\Omega^{(k)} - \Omega^{(k+1)}| < \varepsilon_3$$

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 ϕ, w, Ω, 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.

<u>Step 1</u>. 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.

<u>Step 2</u>. At each point of S_h, set

$$\phi = 0 \; . \tag{3.7}$$

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

$$\phi(1 - \Delta r, \alpha) = \frac{\phi(1 - 2\Delta r, \alpha)}{4} \; . \tag{3.8}$$

On the remaining points of R_h, write down (2.20) with Ω_0 replaced by $\Omega_0^{(k)}$. Solve the linear algebraic system so generated by SOR [4] with over-relaxation factor r_ϕ and denote the solution by $\bar{\phi}^{(k+1)}$. Then, define $\phi^{(k+1)}$ on $R_h \cup S_h$ by the smoothing formula

$$(3.9) \qquad \phi^{(k+1)} = \xi_1 \phi^{(k)} + (1 - \xi_1) \bar{\phi}^{(k+1)}, \quad 0 \le \xi_1 \le 1.$$

Step 3. At each point of S_h for which $r = 1$, set $w = 0$. At the origin write down (2.10) with each ϕ_i replaced by the known value $\phi_i^{(k+1)}$ given by (3.9), with ϕ_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 ϕ_i replaced by $\phi_i^{(k+1)}$. On the remaining points of S_h, write down (2.21) with ϕ_i replaced by $\phi_i^{(k+1)}$, with ϕ_4 replaced by $-\phi_2$ and w_4 replaced by w_2 between O and P_1, and with ϕ_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_w as over-relaxation factor, and denote the solution by $\bar{w}^{(k+1)}$. Then, define $w^{(k+1)}$ on $R_h \cup S_h$ by

$$(3.10) \qquad w^{(k+1)} = \xi_2 w^{(k)} + (1 - \xi_2) \bar{w}^{(k+1)}, \quad 0 \le \xi_2 \le 1.$$

Step 4. At each point of S_h for which $r = 1$, set

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

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

$$\Omega^{(k+1)} = \xi_3 \Omega^{(k)} + (1 - \xi_3)\bar{\Omega}^{(k+1)}, \quad 0 \le \xi_3 \le 1.$$

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 ϕ_i replaced by $\phi_i^{(k+1)}$, with w_i replaced by $w_i^{(k+1)}$, and with

Ω 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_Ω. Denote the solution by $\bar{\Omega}^{(k+1)}$. Finally, define $\Omega^{(k+1)}$ on the point set not included in Step 4 by

$$(3.12) \qquad \Omega^{(k+1)} = \xi_4 \Omega^{(k)} + (1 - \xi_4)\bar{\Omega}^{(k+1)}, \quad 0 \le \xi_4 \le 1 .$$

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 ϕ, w, and Ω were set at $\frac{1}{20}\varepsilon_i$, $i = 1,2,3$, respectively. Graphs of constant-ϕ 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-ϕ 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

D	ε_1	ε_2	ε_3	r_ϕ	r_w	r_Ω	ξ_1	ξ_2	ξ_3	ξ_4	Number Outer Iterations for Convergence
10	10^{-5}	10^{-3}	10^{-4}	1.5	1.8	1.5	0.1	0.1	0.1	0.1	8
100	$2\cdot10^{-4}$	$5\cdot10^{-3}$	$5\cdot10^{-3}$	1.5	1.7	1.5	0.1	0.1	0.1	0.1	8
250	$2\cdot10^{-3}$	$2\cdot10^{-2}$	$4\cdot10^{-2}$	1.5	1.8 on iteration #1 only, then 1.5	1.5	0.1	0.1	0.1	0.1	9
500	$4\cdot10^{-3}$	$4\cdot10^{-2}$	$8\cdot10^{-2}$	1.5	1.5	1.5	0.1	0.1	0.1	0.1	13
1000	$5\cdot10^{-3}$	$5\cdot10^{-2}$	$17\cdot10^{-2}$	1.5	1.5	1.5	0.5	0.1	0.1	0.5	21
2000	$7\cdot10^{-3}$	$8\cdot10^{-2}$	$3\cdot10^{-1}$	1.5	1.5	1.3	0.5	0.1	0.1	0.5	25
5000	10^{-2}	$15\cdot10^{-2}$	$6\cdot10^{-1}$	1.5	1.5	1.3	0.3	0.1	0.1	0.7	25

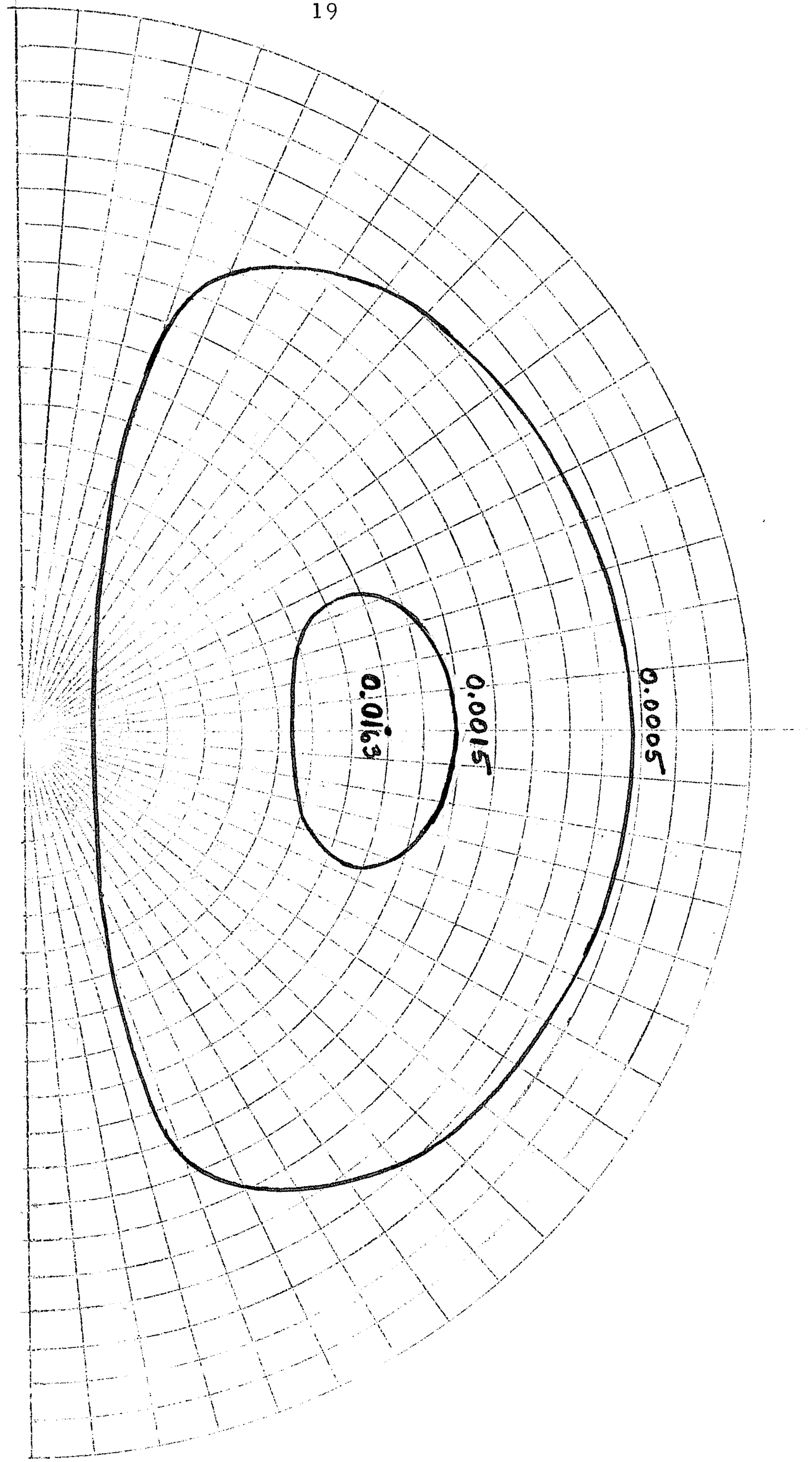

FIG. 3 - CONSTANT-ϕ CURVES FOR D = 10

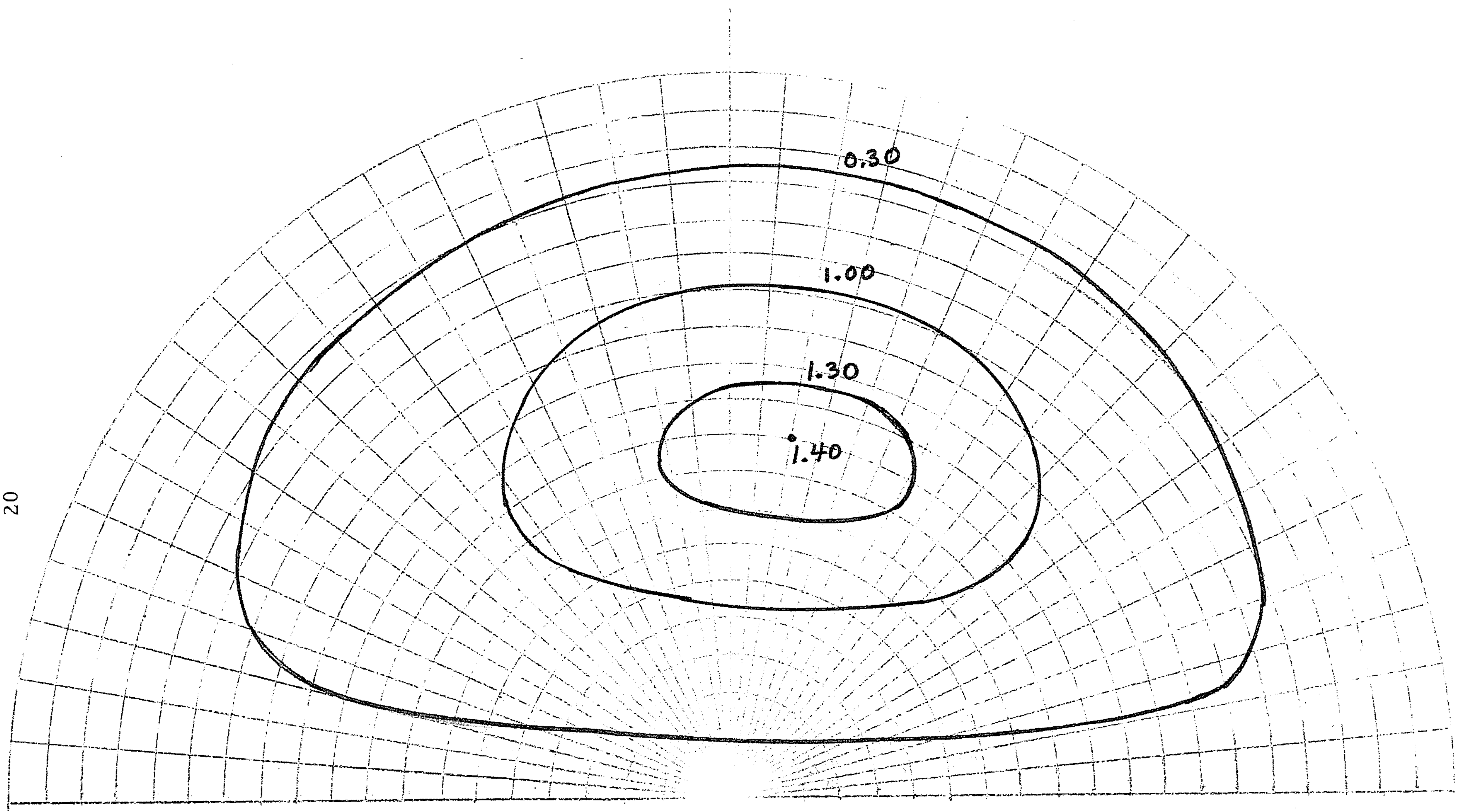

FIGURE 4. - CONSTANT-ϕ CURVES FOR D = 100

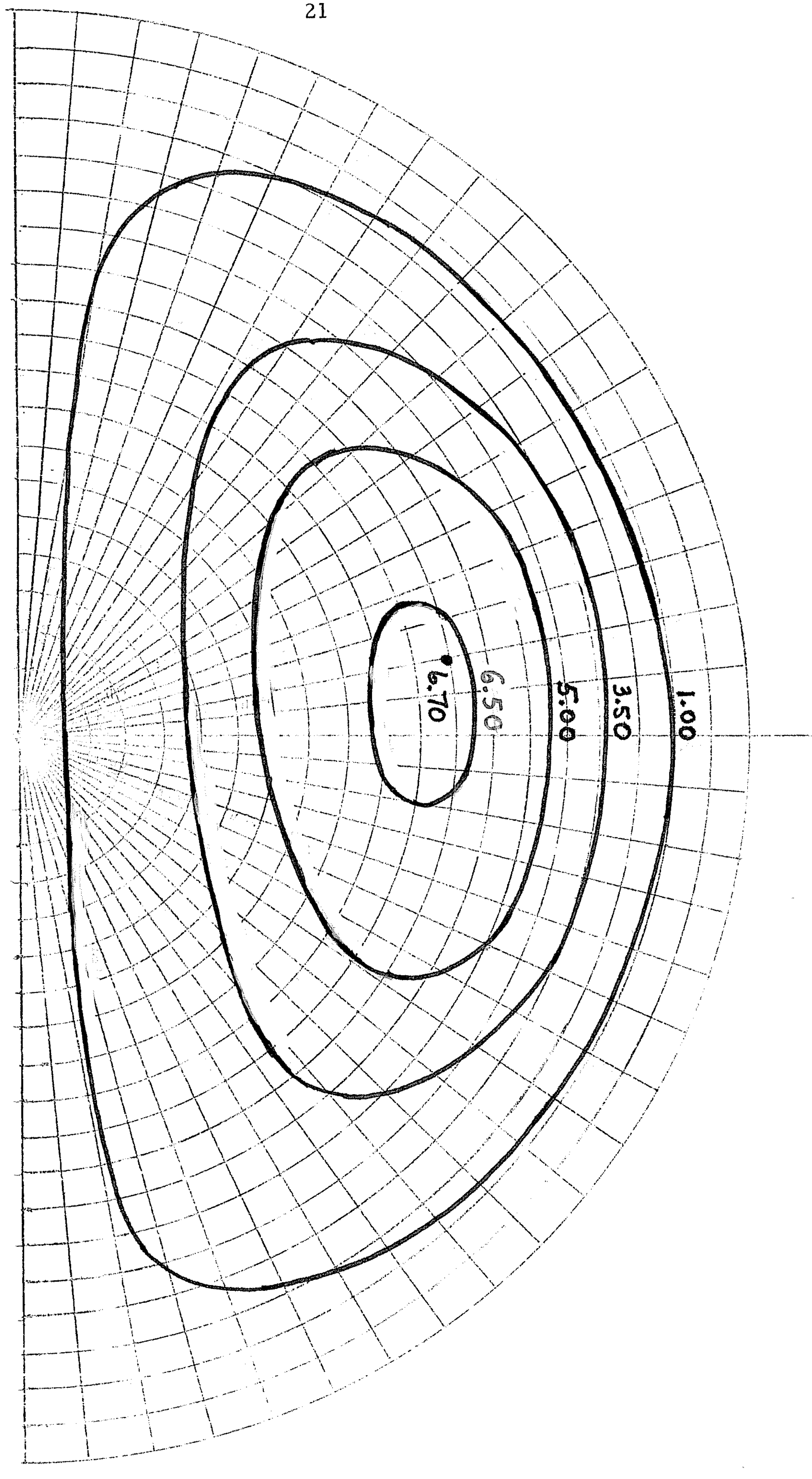

FIG. 5 - CONSTANT-ϕ CURVES FOR D=500

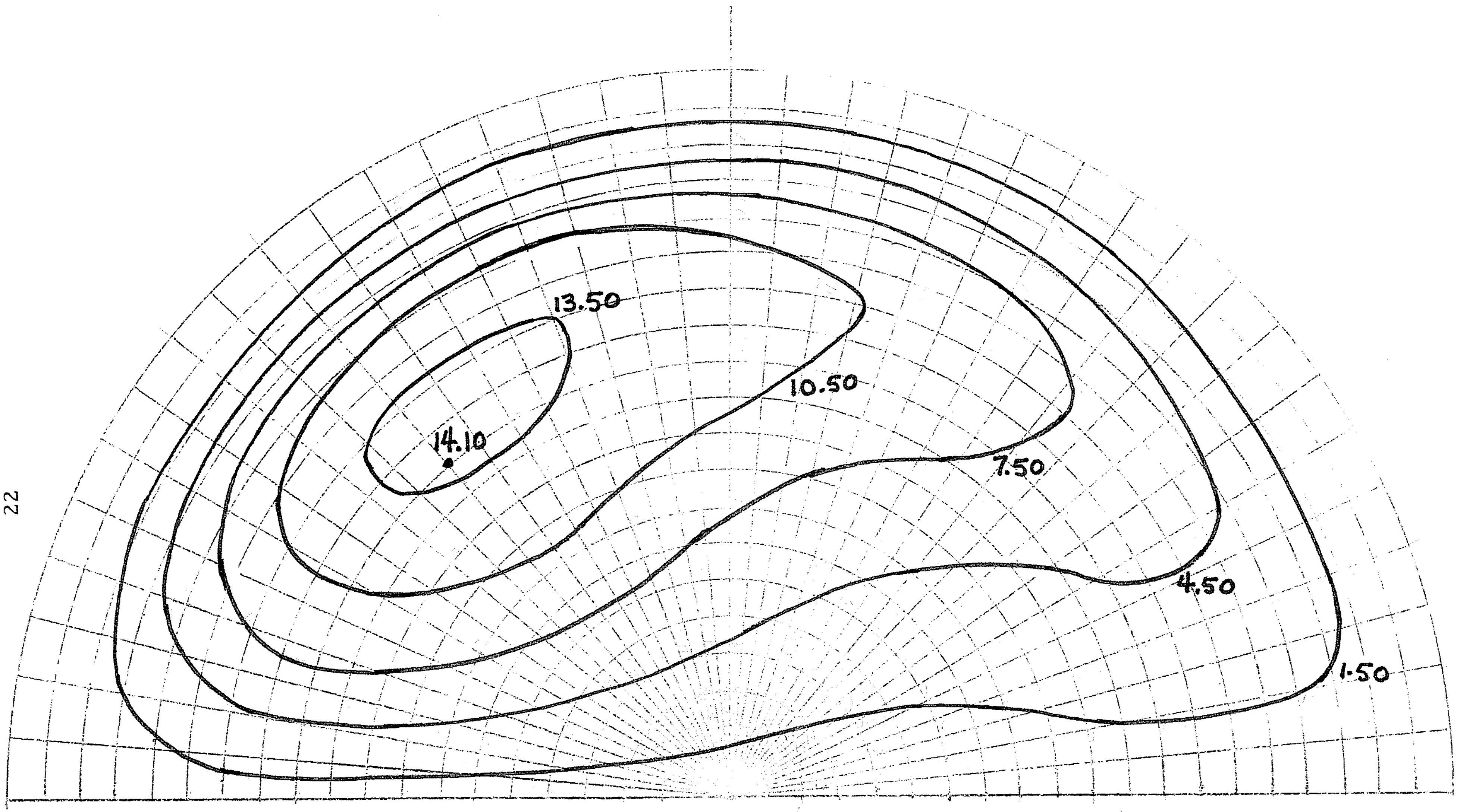

FIG. 6 - CONSTANT-ϕ CURVES FOR D = 2000

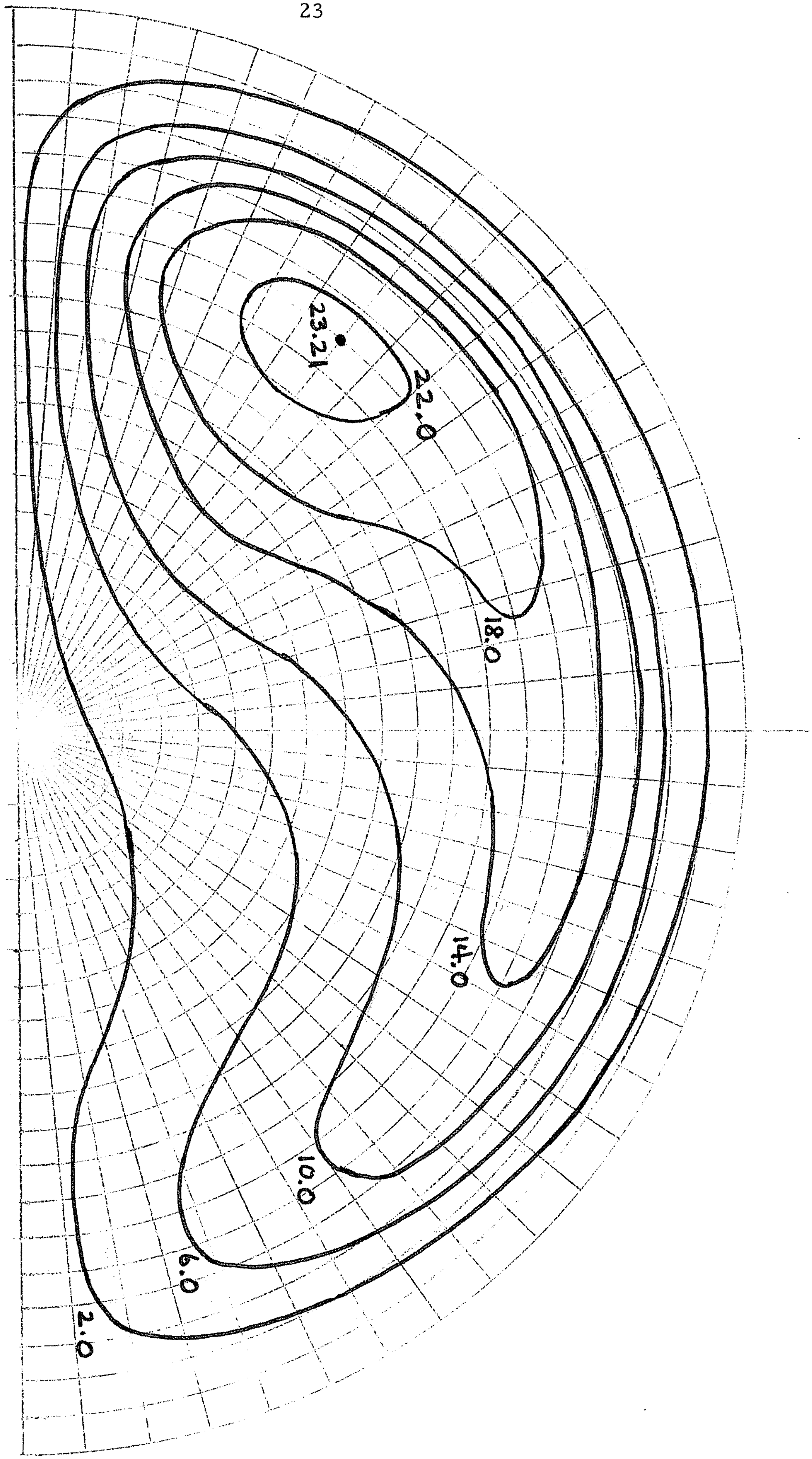

FIGURE 7 - CONSTANT-ϕ CURVES FOR D = 5000

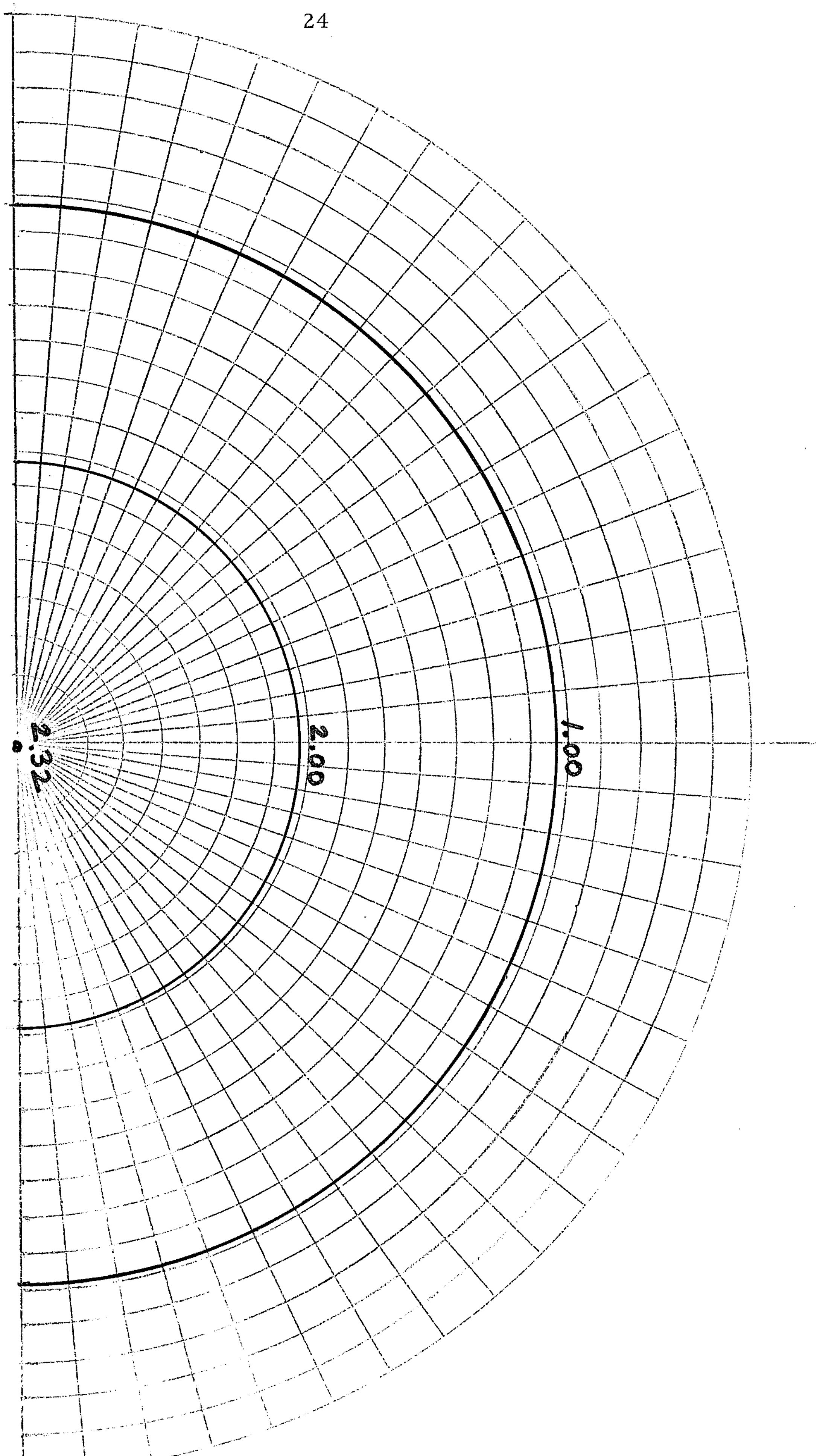

FIGURE 8 - CONSTANT-W CURVES FOR D = 10

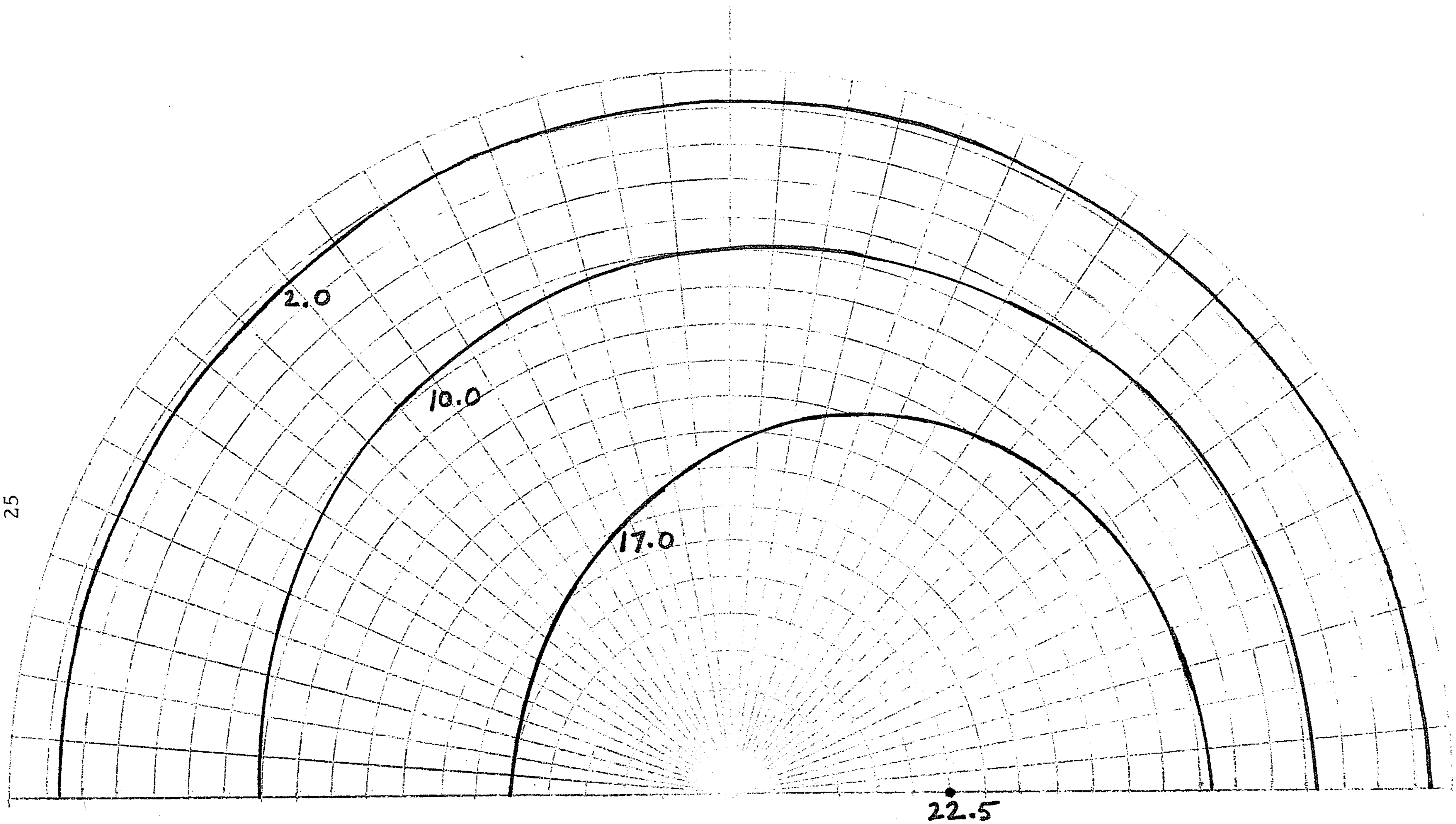

FIGURE 9 - CONSTANT-w CURVES FOR D = 100

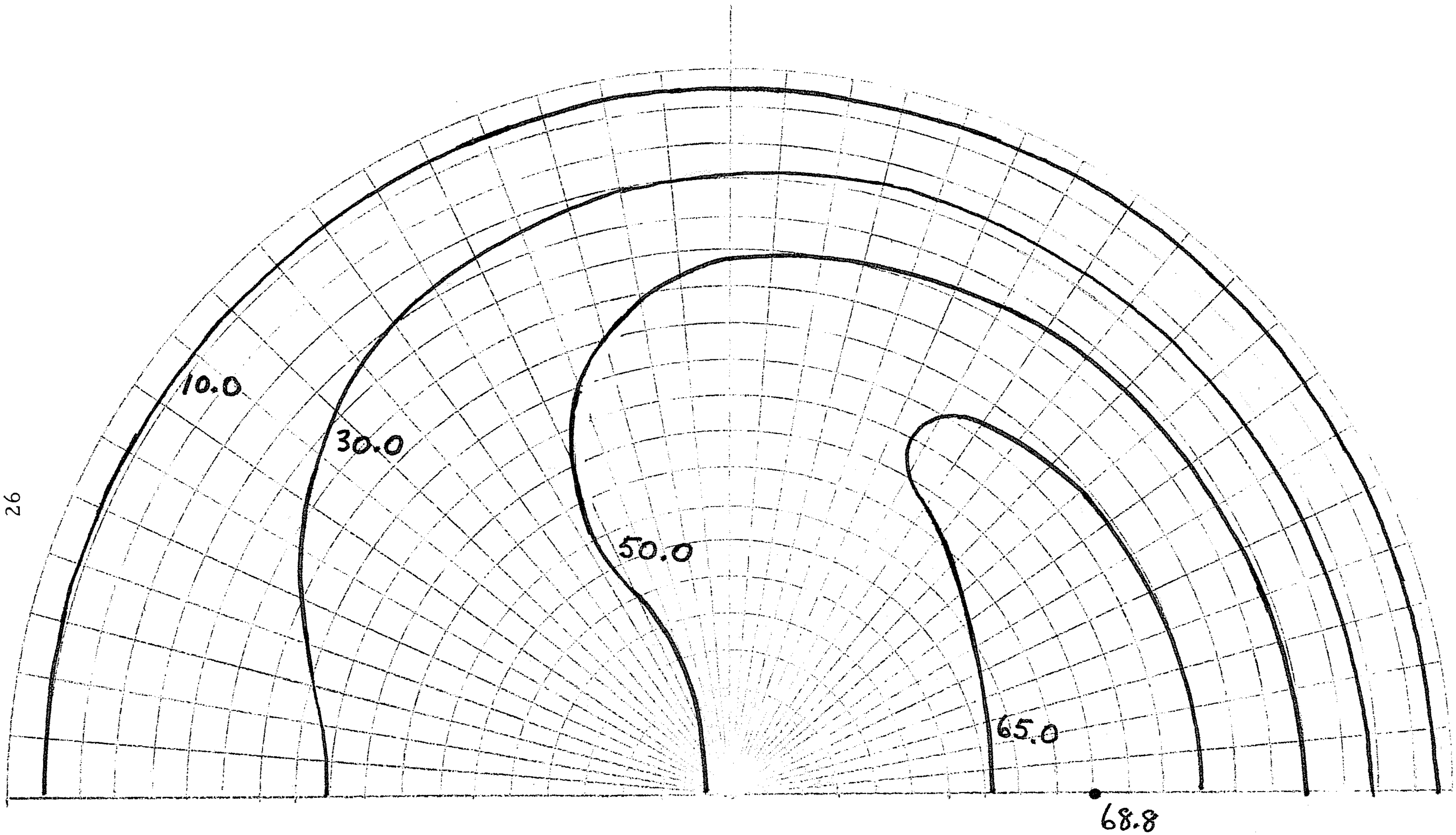

FIGURE 10 - CONSTANT-w CURVES FOR D = 500

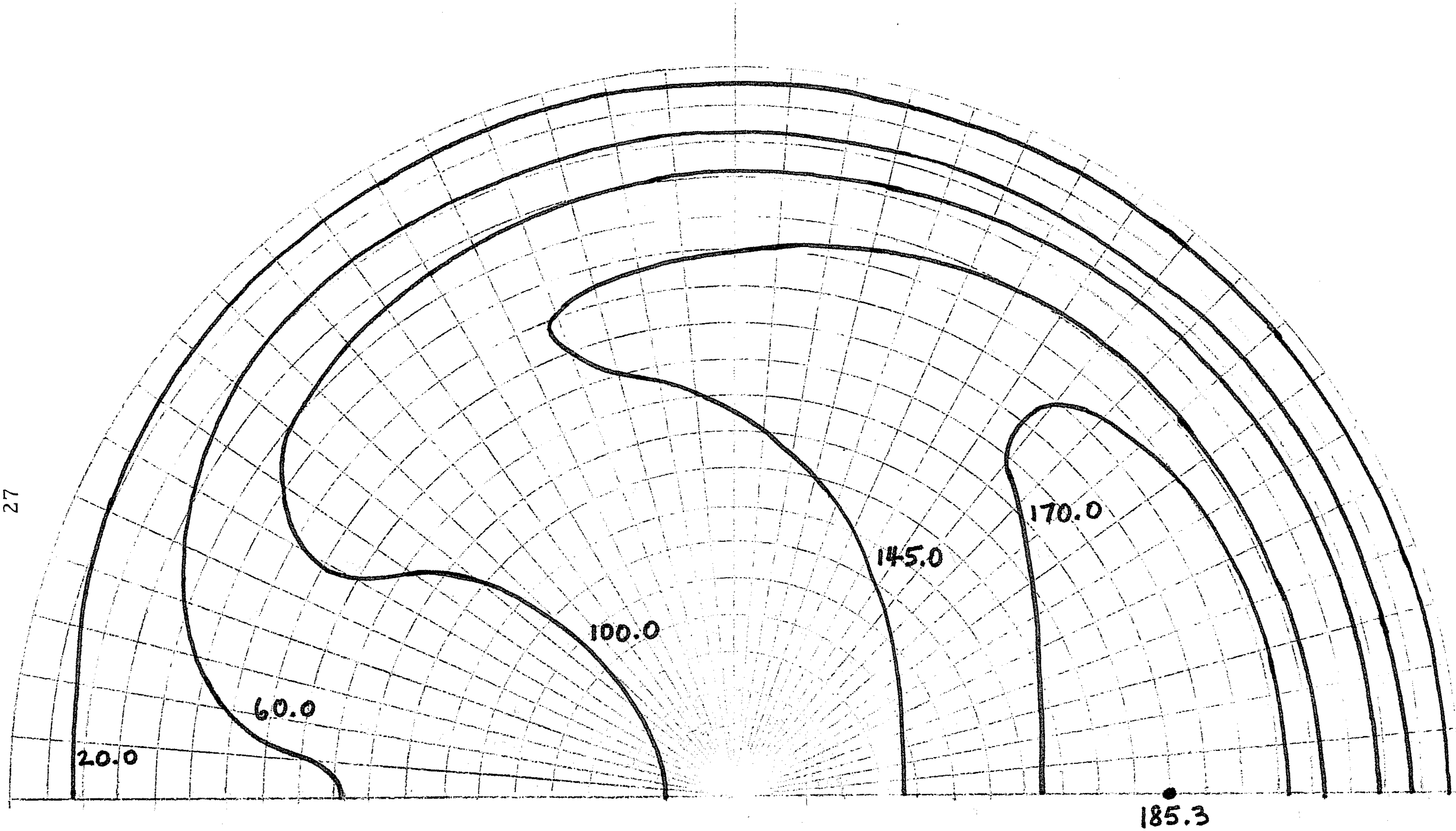

FIGURE 11 - CONSTANT-w CURVES FOR D = 2000

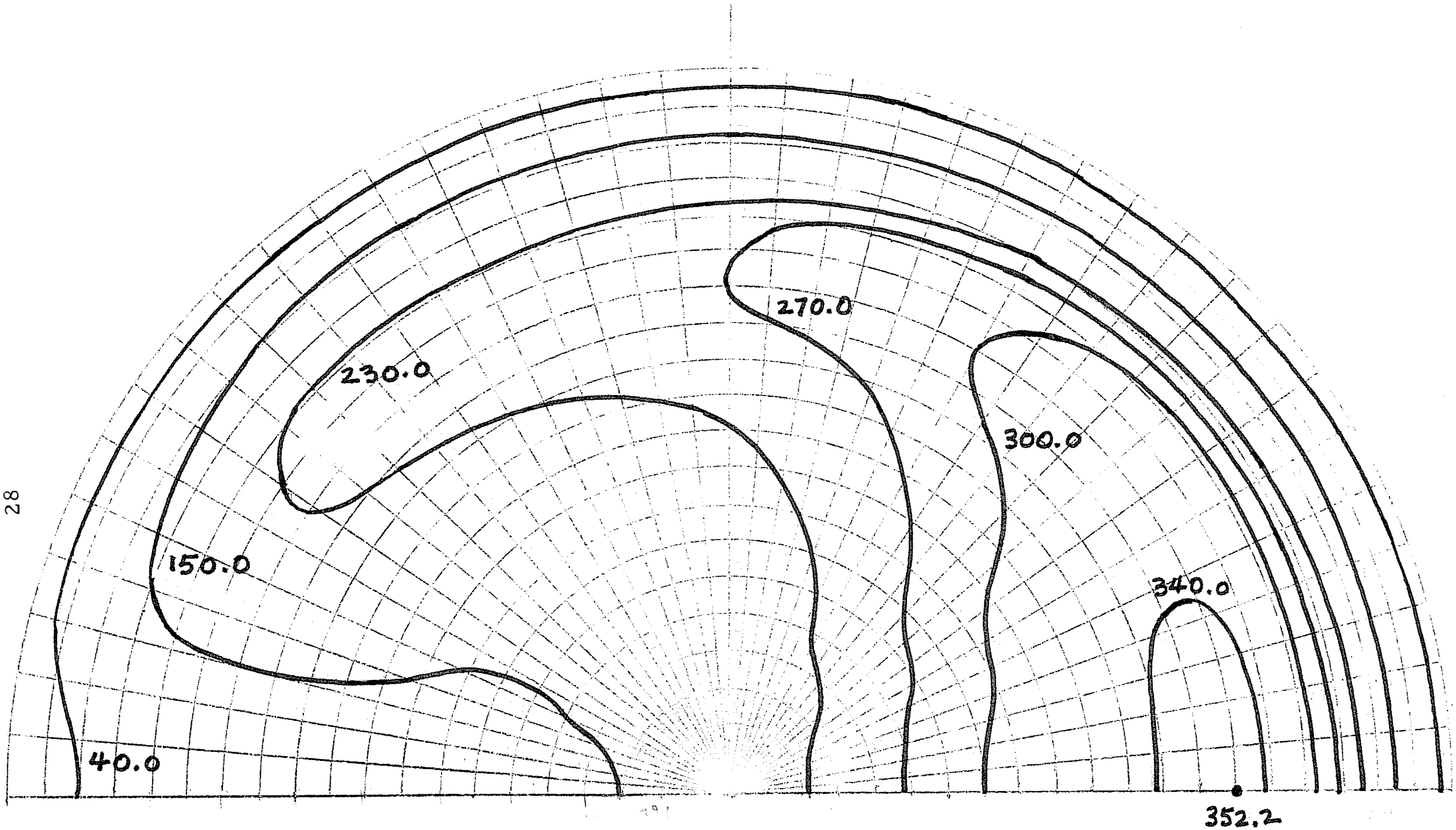

FIGURE 12 - CONSTANT-w CURVES FOR D = 5000

References

1. W. R. Dean, "Note on the motion of fluid in a curved pipe," Philos. Mag., 4, 1927, p. 208.

2. __________, "The stream-line motion of fluid in a curved pipe," Philos. Mag., 5, 1928, p. 673.

3. J. Eustice, "Experiments on stream-line motion in curved pipes," Proc. Roy. Soc. A, LXXXV, 1911, p. 119.

4. D. Greenspan, Lectures on the Numerical Solution of Linear, Singular, and Nonlinear Differential Equations, Prentice-Hall, Englewood Cliffs, N. J., 1968, p. 2.

5. __________, "Numerical studies of prototype cavity flow problems," The Computer J., 12, 1969, p. 89.

6. D. J. McConalogue and R. S. Srivastava, "Motion of a fluid in a curved tube," Proc. Roy. Soc. A, 307, 1968, p. 37.

7. A. B. Schubert, "FORTRAN program for secondary flow," TR 155, Dept. of Comp. Sci., Univ. Wis., Madison, 1972.

8. G. I. Taylor, "The criterion for turbulence in curved pipes," Proc. Roy. Soc. A, 124, 1929, p. 243.

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

```
C      TUBEFLOW     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=10,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)
C                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                     I = 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 SOR ITERATION
C                     K = 3 IMPLIES A VALUE AT THE CURRENT SOR AND OUTER
C                           ITERATION.
C          W(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),
      * XI(4),RHO(3),EPS(3),SA(NAP1),COSA(NAP1),RINV(NRP1),RINV2(NRP1),
      * RHOC(3),B(NRP1,5),C(NRP1,NAP1),EPPS(3),XIC(4),E(NRP1,NAP1,6),
      * EE(NRP1),EF(NRP1),CA(NRP1,NAP1),DELA(NR),CO(NR),DOR(5)

       DATA PI/3.14159265/,MAXSOR,MAXOUT/250,60/,NOR/3/,(DOR(I),I=1,3)
      */.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         NOR = NUMBER OF DIFFERENT OVER-RELAXATION FACTORS TO BE TRIED
C               IN ANY DATA CASE.
C      DOR(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 IMPLIES STARTING VALUES WILL BE SET = 0 BY
C                           PROGRAM.
C               ISTART.NE.0 IMPLIES STARTING ITERATES WILL BE READ FROM
```

```
C                                CARDS IN THE FORMAT (8E10.5).
C      TEST FOR NECESSITY OF READING STARTING OUTER ITERATES FOR PHI, W,
C      AND OMEGA.
C      DSTART = VALUE OF PARAMETER D FOR WHICH THE STARTING VALUES
C      ARE A SOLUTION.
       IF(ISTART.NE.0) READ(5,33) DSTART,((PHI(I,J,4),J=1,NAP1),I=1,NRP1)
      *,((W(I,J,4),J=1,NAP1),I=1,NRP1),((OMEGA(I,J,4),J=1,NAP1),I=1,NRP1)
       KRP1=NRP1
C      DR = GRID SPACING IN  RADIAL DIRECTION.
C      DA = GRID SPACING IN ANGULAR DIRECTION.
       DR=1./NR
       DA=PI/NA
       DAH=.5*DA
       DRH=.5*DR
       DRDAM=DR*DA
       DRDA=DR/DA
       DADR=DA/DR
C      COMPUTE ELEMENTS OF AREA, DELA(I),I=1,....,NR, IN RADIAL DIRECTION
C      FOR USE IN FLUX CALCULATION.
       DO 1 I=1,NR
     1 DELA(I)=(2*I-1)*DRH*DRDAM
C      ON THE GRID DEFINED BY DR AND DA, COMPUTE DISCRETE FUNCTIONS
C      DEPENDENT ONLY ON RADIUS AND ANGLE.
       SA(1)=0.
       DO 2 J=2,NA
       SA(J)=SIN((J-1)*DA)*DAH
     2 COSA(J)=COS((J-1)*DA)
       SA(NAP1)=0.
       COSA(NAP1)=-1.
       RINV(NRP1)=1.
       RINV2(NRP1)=1.
       DO 3 I=2,NR
       RINV(I)=1./((I-1)*DR)
       DRRH=DRH*RINV(I)
       RINV2(I)=RINV(I)**2
       DO 3 J=2,NA
     3 CA(I,J)=DRRH*COSA(J)
C      COMPUTE SOR COEFFICIENTS FOR PHI.
       DO 4 I=2,NR
       BC=2.*(DADR+DRDA*RINV2(I))+DA*RINV(I)
       B(I,1)=(DADR+DA*RINV(I))/BC
       B(I,2)=DRDA*RINV2(I)/BC
       B(I,3)=DADR/BC
       B(I,4)=B(I,2)
```

```
    4 B(I,5)=DRDAM/30
C     READ OUTER ITERATION SMOOTHING PARAMETERS (XI), OVER-RELAXATION
C     FACTORS (RHO), OUTER ITERATION CONVERGENCE TOLERANCES (EPS),
C     PARAMETER RELATED TO REYNOLDS NO. (D), AND INPUT/OUTPUT CONTROL
C     VARIABLES DEFINED AS FOLLOWS.
C     ISAVE.NE.0 IMPLIES SAVE THE SOLUTION FOR THIS CASE (IF OBTAINED)
C                IN MEMORY FOR USE AS STARTING VALUES FOR A SUCCEEDING
C                CASE WITH A DIFFERENT VALUE OF D.
C     ISAVE.EQ.0 IMPLIES DO NOT DO THE ABOVE.
C      IUSE.NE.0 IMPLIES TAKE STARTING VALUES FOR OUTER ITERATION FROM
C                MEMORY.
C      IUSE.EQ.0 IMPLIES INITIALIZE OUTER ITERATES TO ZERO.
C      IPCH.NE.0 IMPLIES PUNCH SOLUTION FOR THIS CASE (IF OBTAINED)
C                OUT ON CARDS IN THE FORMAT (8E10.5).
C      IPCH.EQ.0 IMPLIES DO NOT PUNCH SOLUTION OBTAINED FOR THIS CASE.
    5 READ(5,99,END=70) XI,RHO,EPS,D,ISAVE,IUSE,IPCH
C     PRINT OUT INPUT PARAMETERS.
      WRITE(6,98) D,RHO,XI,EPS
C     COMPUTE PARAMETERS DEPENDENT ON THE INPUT PARAMETER D, INCLUDING
C     THE COEFFICIENTS, CO(I),I=1,....,NR, IN THE NUMERICAL INTEGRATION
C     FOR THE FLUX.
      DDRDAM=D*DRDAM
      DDR2=D*DR**2
      CON=4./(PI*D)
      CO(1)=16.*DELA(1)/(3.*PI*D)
      DO 105 I=2,NR
  105 CO(I)=CON*DELA(I)
C     COMPUTE COMPLEMENTS (RELATIVE TO 1) OF SMOOTHING PARAMETERS AND
C     OVER-RELAXATION FACTORS.
C     ALSO COMPUTE SOR CONVERGENCE TOLERANCES, EPPS(I),I=1,2,3, AS
C     FUNCTIONS OF OUTER ITERATION TOLERANCES, EPS(I),I=1,2,3.
      DO 6 I=1,3
      XIC(I)=1.-XI(I)
      RHOC(I)=1.-RHO(I)
    6 EPPS(I)=.05*EPS(I)
      XIC(4)=1.-XI(4)
C     TEST WHETHER INITIAL OUTER ITERATES ARE TO BE OBTAINED FROM
C     MEMORY, HAVING BEEN PLACED THERE AS THE SOLUTION FOR A PREVIOUS
C     VALUE OF D EITHER BY CARD INPUT OR BY A PREVIOUS COMPUTATION
C     THIS RUN.
      IF(IUSE.EQ.0) GO TO 7
      DO 106 I=1,NRP1
      DO 106 J=1,NAP1
      PHI(I,J,3)=PHI(I,J,4)
```

```
      W(I,J,3)=W(I,J,4)
  106 OMEGA(I,J,3)=OMEGA(I,J,4)
      WRITE(6,87) DSTART
      GO TO 9

C     INITIALIZE OUTER ITERATES TO ZERO IF NEITHER INPUT NOR COMPUTED
C     PREVIOUSLY THIS RUN.

    7 DO 8 I=1,NRP1
      DO 8 J=1,NAP1
      PHI(I,J,3)=0.
      W(I,J,3)=0.
    8 OMEGA(I,J,3)=0.

C     INITIALIZE OUTER ITERATION COUNTER (IOUT) AND COUNTERS OF NUMBER
C     OF OVER-RELAXATION FACTORS USED FOR W AND OMEGA (IRW,IRO, RESP.)
C     FOR THIS DATA CASE.

    9 IOUT=0
      IRW=0
      IRO=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(6,92) IOUT
      ICV=0

C     UPDATE PREVIOUS OUTER ITERATES.

      DO 12 I=1,NRP1
      DO 12 J=1,NAP1
      PHI(I,J,1)=PHI(I,J,3)
      W(I,J,1)=W(I,J,3)
   12 OMEGA(I,J,1)=OMEGA(I,J,3)

C     COMPUTE CONSTANT SOR COEFFICIENT FOR PHI.

      DO 13 I=2,NR
      DO 13 J=2,NA
   13 C(I,J)=B(I,5)*OMEGA(I,J,1)

C     INITIALIZE SOR ITERATION COUNTER FOR PHI.

      ISOR=0
```

```
C     TEST FOR MAXIMUM NUMBER OF SOR ITERATIONS FOR PHI.

   14 IF(ISOR.GE.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 16 I=2,NR
      DO 16 J=2,NA
   16 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=0

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

      DO 18 I=2,NRM1
      DO 18 J=2,NA
      PHI(I,J,3)=RHOC(1)*PHI(I,J,2)+RHO(1)*(B(I,1)*PHI(I+1,J,2)
     $ +B(I,2)*PHI(I,J+1,2)+B(I,3)*PHI(I-1,J,3)+B(I,4)*PHI(I,J-1,3)
     $ +C(I,J))
      IF(ABS(PHI(I,J,2)-PHI(I,J,3)).GT.EPPS(1)) ICONV=1
   18 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(ABS(PHI(NR,J,2)-PHI(NR,J,3)).GT.EPPS(1)) ICONV=1
   19 CONTINUE

C     TEST FOR CONVERGENCE OF SOR ITERATION FOR PHI.

      IF(ICONV.NE.0) GO TO 14

C     SOR CONVERGENCE ATTAINED FOR PHI. SMOOTH OUTER ITERATE.

      DO 20 I=2,NR
      DO 20 J=2,NA
      PHI(I,J,3)=XI(1)*PHI(I,J,1)+XIC(1)*PHI(I,J,3)
      IF(ABS(PHI(I,J,1)-PHI(I,J,3)).GT.EPS(1)) ICV=1
   20 CONTINUE
```

```
C     PRINT OUT SMOOTHED CURRENT OUTER ITERATE FOR PHI.

      CALL OUTPUT('PHI',ISOR,PHI(1,1,3))

C     COMPUTE COEFFICIENTS FOR SOR ITERATION FOR W...
C     ...FIRST COMPUTE COEFFICIENTS AT THE ORIGIN

      E0=4.+ABS(PHI(2,NAH,3))
      E1=(1.-AMIN1(PHI(2,NAH,3),0.))/E0
      E2=2./E0
      E3=(1.+AMAX1(PHI(2,NAH,3),0.))/E0
      E4=DDR2/E0

C     ...NEXT COMPUTE COEFFICIENTS FOR REMAINDER OF REGION...
C           FIRST FOR RAYS ALONG ALPHA = 0 AND ALPHA = PI, EXCLUDING
C           R = 0 AND R = 1.

      DO 23 I=2,NR
      DELTA1=DA-PHI(I,2,3)
      DELTA2=DA+PHI(I,NA,3)
      EE(I)=2.*(DADR+DRDA*RINV2(I))
      EE1=EE(I)+RINV(I)*ABS(DELTA1)
      EE2=EE(I)+RINV(I)*ABS(DELTA2)
      E(I,1,1)=(DADR+RINV(I)*AMAX1(DELTA1,0.))/EE1
      E(I,NAP1,1)=(DADR+RINV(I)*AMAX1(DELTA2,0.))/EE2
      EF(I)=DRDA*RINV2(I)
      E(I,1,2)=EF(I)/EE1
      E(I,NAP1,2)=EF(I)/EE2
      E(I,1,3)=(DADR-RINV(I)*AMIN1(DELTA1,0.))/EE1
      E(I,NAP1,3)=(DADR-RINV(I)*AMIN1(DELTA2,0.))/EE2
      E(I,1,4)=E(I,1,2)
      E(I,NAP1,4)=E(I,NAP1,2)
      E(I,1,5)=DDRDAM/EE1
   23 E(I,NAP1,5)=DDRDAM/EE2

C           THEN ON INTERIOR OF REGION.

      DO 25 I=2,NR
      DO 25 J=2,NA
      GAMMA=.5*(PHI(I+1,J,3)-PHI(I-1,J,3))
      DELTA=DA-.5*(PHI(I,J+1,3)-PHI(I,J-1,3))
      E(I,J,6)=EE(I)+RINV(I)*(ABS(GAMMA)+ABS(DELTA))
      E(I,J,1)=(DADR+RINV(I)*AMAX1(DELTA,0.))/E(I,J,6)
      E(I,J,2)=(EF(I)+RINV(I)*AMAX1(GAMMA,0.))/E(I,J,6)
      E(I,J,3)=(DADR-RINV(I)*AMIN1(DELTA,0.))/E(I,J,6)
      E(I,J,4)=(EF(I)-RINV(I)*AMIN1(GAMMA,0.))/E(I,J,6)
   25 E(I,J,5)=DDRDAM/E(I,J,6)

C     INITIALIZE SOR ITERATION COUNTER FOR W.

   26 ISOR=0

C     TEST FOR MAXIMUM NUMBER OF SOR ITERATIONS.
```

```
   27 IF(ISOR.GE.MAXSOR) GO TO 36
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(1,1,2)=W(1,1,3)
C     ...THEN ON REMAINDER OF REGION, EXCLUDING R = 1.
      DO 29 I=2,NR
      DO 29 J=1,NAP1
   29 W(I,J,2)=W(I,J,3)
C     SET SOR CONVERGENCE INDICATOR (ICONV) TO ZERO.
      ICONV=0
C     COMPUTE CURRENT SOR ITERATE FOR W...
C     ...FIRST AT THE ORIGIN
      W(1,1,3)=RHOC(2)*W(1,1,2)     +RHO(2)*(E1*W(2,1,2)+E2*W(2,NAH,2)
     $ +E3*W(2,NAP1,2)+E4)
C     (SET VALUES OF W FOR R = 0 AND ALL ANGLES ALPHA(I),I=1,...,NA+1,
C     EQUAL TO VALUE OF W AT ORIGIN FOR CONVENIENCE IN SOR COMPUTATION.)
      DO 30 J=2,NAP1
   30 W(1,J,3)=W(1,1,3)
      IF(ABS(W(1,1,2)-W(1,1,3)).GT.EPPS(2)) ICONV=1
C     ...NEXT ALONG RAY ALPHA = 0, EXCLUDING R = 0 AND R = 1.
      DO 32 I=2,NR
      W(I,1,3)=RHOC(2)*W(I,1,2)+RHO(2)*(E(I,1,1)*W(I+1,1,2)+E(I,1,2)*2.*
     $ W(I,2,2)+E(I,1,3)*W(I-1,1,3)+E(I,1,5))
      IF(ABS(W(I,1,3)-W(I,1,2)).GT.EPPS(2)) ICONV=1
   32 CONTINUE
C     ...NEXT ON INTERIOR OF REGION
      DO 33 I=2,NR
      DO 33 J=2,NA
      W(I,J,3)=RHOC(2)*W(I,J,2)+RHO(2)*(E(I,J,1)*W(I+1,J,2)+E(I,J,2)*
     $ W(I,J+1,2)+E(I,J,3)*W(I-1,J,3)+E(I,J,4)*W(I,J-1,3)+E(I,J,5))
      IF(ABS(W(I,J,2)-W(I,J,3)).GT.EPPS(2)) ICONV=1
   33 CONTINUE
C     ...FINALLY ALONG RAY ALPHA = PI, EXCLUDING R = 0 AND R = 1.
      DO 34 I=2,NR
```

```
      W(I,NAP1,3)=RHOC(2)*W(I,NAP1,2)+RHO(2)*(E(I,NAP1,1)*W(I+1,NAP1,2)+
     $ E(I,NAP1,3)*W(I-1,NAP1,3)+2.*E(I,NAP1,4)*W(I,NA,3)+E(I,NAP1,5))
      IF(ABS(W(I,NAP1,2)-W(I,NAP1,3)).GT.EPPS(2)) ICONV=1
   34 CONTINUE

C     TEST FOR CONVERGENCE OF SOR ITERATION FOR W.

      IF(ICONV.NE.0) GO TO 27

C     SOR CONVERGENCE ATTAINED FOR W. SMOOTH OUTER ITERATE...
C     ...FIRST AT THE ORIGIN

      W(1,1,3)=XI(2)*W(1,1,1)+XIC(2)*W(1,1,3)

C     (SET VALUES OF W FOR R = 0 AND ALL DISCRETE ANGLES EQUAL TO
C     SMOOTHED VALUE OF W AT ORIGIN.)

      DO 134 J=2,NAP1
  134 W(1,J,3)=W(1,1,3)
      IF(ABS(W(1,1,1)-W(1,1,3)).GT.EPS(2)) ICV=1

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

      DO 35 I=2,NR
      DO 35 J=1,NAP1
      W(I,J,3)=XI(2)*W(I,J,1)+XIC(2)*W(I,J,3)
      IF(ABS(W(I,J,1)-W(I,J,3)).GT.EPS(2)) ICV=1
   35 CONTINUE

C     PRINT OUT SMOOTHED CURRENT OUTER ITERATE FOR W.

      CALL OUTPUT('W',ISOR,W(1,1,3))

C     COMPUTE AND SMOOTH OMEGA ON BOUNDARY R = 1.

      DO 37 J=2,NA
      OMEGA(NRP1,J,3)=XI(3)*OMEGA(NRP1,J,1)-XIC(3)*2.*RINV2(2)*PHI(NR,
     $ J,3)
      IF(ABS(OMEGA(NRP1,J,1)-OMEGA(NRP1,J,3)).GT.EPS(3)) ICV=1
   37 CONTINUE

C     COMPUTE CONSTANT COEFFICIENT FOR SOR ITERATION FOR OMEGA.

      DO 40 I=2,NR
      DO 40 J=2,NA
   40 E(I,J,5)=-W(I,J,3)*(SA(J)*(W(I+1,J,3)-W(I-1,J,3))+CA(I,J)*(W(I,
     $ J+1,3)-W(I,J-1,3)))/E(I,J,6)

C     INITIALIZE COUNTER OF SOR ITERATION NUMBER FOR OMEGA.

   41 ISOR=0

C     TEST FOR MAXIMUM NUMBER OF SOR ITERATIONS.
```

```
   42 IF(ISOR.GE.MAXSOR) GO TO 57
C      UPDATE SOR ITERATION COUNTER IF MAXIMUM NUMBER HAS NOT BEEN
C      ACHIEVED.
       ISOR=ISOR+1
C      UPDATE PREVIOUS SOR ITERATES FOR OMEGA ON INTERIOR.
       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(3)*OMEGA(I,J,2)+RHO(3)*(E(I,J,1)*OMEGA(I+1,J,2)
      $ +E(I,J,2)*OMEGA(I,J+1,2)+E(I,J,3)*OMEGA(I-1,J,3)+E(I,J,4)
      $ *OMEGA(I,J-1,3)+E(I,J,6))
       IF(ABS(OMEGA(I,J,2)-OMEGA(I,J,3)).GT.EPPS(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 48 I=2,NR
       DO 48 J=2,NA
       OMEGA(I,J,3)=XI(4)*OMEGA(I,J,1)+XIC(4)*OMEGA(I,J,3)
       IF(ABS(OMEGA(I,J,1)-OMEGA(I,J,3)).GT.EPS(3)) ICV=1
   48  CONTINUE
C      PRINT OUT SMOOTHED CURRENT OUTER ITERATE FOR OMEGA.
       CALL OUTPUT('OMEGA',ISOR,OMEGA(1,1,3))
C      TEST FOR CONVERGENCE OF ALL OUTER ITERATES.
       IF(ICV.NE.0) GO TO 10
C      PRINT MESSAGE INDICATING CONVERGENCE OF OUTER ITERATION.
       WRITE(6,97)
C      PUNCH SOLUTION (AND VALUE OF D) OUT ON CARDS IF INPUT CONTROL
C      PARAMETER IPCH SO DICTATES, AND PRINT MESSAGE INDICATING THAT
C      THIS PUNCHING WAS DONE.
```

```
      IF(IPCH.NE.0)PUNCH88,D,((PHI(I,J,3),J=1,NAP1),I=1,NRP1),((W(I,J,3)
     * ,J=1,NAP1),I=1,NRP1),((OMEGA(I,J,3),J=1,NAP1),I=1,NRP1)
      IF(IPCH.NE.0) WRITE(6,86)

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

      QR=0.
      DO 49 J=1,NA
   49 QR=QR+W(1,J,3)+W(2,J,3)+W(2,J+1,3)
      QR=QR*CO(1)
      DO 249 I=2,NR
      S=0.
      DO 149 J=1,NA
  149 S=S+W(I,J,3)+W(I+1,J,3)+W(I,J+1,3)+W(I+1,J+1,3)
  249 QR=QR+CO(I)*S

C     PRINT OUT VALUE OF FLUX RATIO JUST COMPUTED.

      WRITE(6,90) 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)
      W(I,J,4)=W(I,J,3)
   50 OMEGA(I,J,4)=OMEGA(I,J,3)

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

      DSTART=D
      GO TO 5

C     CONVERGENCE FAILURE MESSAGES

C     PHI ITERATION FAILED.  READ NEXT DATA CASE.

   55 WRITE(6,96)
      GO TO 5

C     W 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(6,95) RHO(2)
      IF(IRW.GE.NOR) GO TO 5
      IRW=IRW+1
      RHO(2)=RHO(2)-DOR(IRW)
```

```
      RHOC(2)=1.-RHO(2)
      DO 156 I=1,NRP1
      DO 156 J=1,NAP1
  156 W(I,J,3)=W(I,J,1)
      GO TO 26

C     OMEGA ITERATION FAILED.  REDUCE O-R FACTOR AND TRY AGAIN, UNLESS
C     THIS HAS ALREADY BEEN DONE THE MAXIMUM NUMBER OF TIMES (NOR),
C     IN WHICH CASE READ THE NEXT DATA CASE.

   57 WRITE(6,94) RHO(3)
      IF(IRO.GE.NOR) GO TO 5
      IRO=IRO+1
      RHO(3)=RHO(3)-DOR(IRO)
      RHOC(3)=1.-RHO(3)
      DO 157 I=1,NRP1
      DO 157 J=1,NAP1
  157 OMEGA(I,J,3)=OMEGA(I,J,1)
      GO TO 41

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

   58 WRITE(6,93)
      GO TO 5

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

   70 STOP

   99 FORMAT(11F5.5,3I5)
   98 FORMAT(1H1 6X 'D =' F6.3//13X 'PHI' 7X 'W' 5X 'OMEGA' // 5X
     * 'RHO =' F6.2,2(3X F6.2)/6X 'XI =' F6.4,3(3X F6.4)/5X 'EPS ='
     * F6.4,2(3X F6.4))
   97 FORMAT('0OUTER ITERATION CONVERGED TO GIVEN TOLERANCES.')
   96 FORMAT('0SOR FOR  PHI FAILED.')
   95 FORMAT('0SOR FOR    W FAILED WITH SOR FACTOR =' F6.2)
   94 FORMAT('0SOR FOR OMEGA FAILED WITH SOR FACTOR =' F6.2)
   93 FORMAT('0OUTER ITERATION FAILED TO CONVERGE.')
   92 FORMAT(//'0OUTER ITERATION' I5//)
   91 FORMAT(1X 11E11.5)
   90 FORMAT('0FLUX RATIO =' F10.5)
   89 FORMAT('0SOR FACTOR FOR W CHANGED TO ' F6.2)
   88 FORMAT(8E10.5)
   87 FORMAT('0INITIAL ITERATE TAKEN FROM SOLUTION FOR D =' F7.0)
   86 FORMAT('0SOLUTION WAS OUTPUT ON PUNCHED CARDS.')

      END


C     THIS ROUTINE PRINTS OUT A DISCRETE FUNCTION IN A RECTANGULAR
```

```
C      FORMAT WHICH IS RELATED IN THE FOLLOWING WAY TO THE POLAR GRID
C      IMPOSED ON THE PHYSICAL REGION.
C         THE TOP LINE OF THE PRINT BLOCK CORRESPONDS TO VALUES OF THE
C         FUNCTION ALONG THE ARC R = 1.
C         THE BOTTOM LINE OF THE PRINT BLOCK CORRESPONDS TO THE VALUE
C         OF THE FUNCTION AT THE ORIGIN (R = 0).

      SUBROUTINE OUTPUT(VAR,ISOR,A)
      PARAMETER NR=10,NA=18,NRP1=NR+1,NAP1=NA+1
      DIMENSION A(NRP1,NAP1)
      DATA APHI,AW,AOMEGA/'PHI   ','W     ','OMEGA '/
   99 FORMAT(1H0 A6,5X I5,2X 'SOR ITERATIONS'/)
   98 FORMAT(1X F6.2,17F7.2,F6.2)
   97 FORMAT(1X F6.1,17F7.1,F6.1)
      KRP1=NRP1
      KAP1=NAP1
      WRITE(6,99) VAR,ISOR
      IF(VAR.NE.APHI) GO TO 8
      DO 5 I=1,NRP1
    5 WRITE(6,98) (A(KRP1-I+1,KAP1-J+1),J=1,KAP1)
      RETURN
    8 DO 10 I=1,NRP1
   10 WRITE(6,97) (A(KRP1-I+1,KAP1-J+1),J=1,KAP1)
      RETURN
      END
```