TECH LIBRARY KAFB, NM
0060853

NASA CONTRACTOR REPORT

NASA CR-1672

NASA CR-1
c. 1

LOAN COPY: RETURN TO
AFWL (WLOL)
KIRTLAND AFB, N MEX

NONLINEAR ACTUATOR DISK THEORY AND FLOW FIELD CALCULATIONS, INCLUDING NONUNIFORM LOADING

by Michael D. Greenberg and Stephen R. Powers

Prepared by
SAGE ACTION, INC.
Ithaca, N. Y. 14850
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1970

TECH LIBRARY KAFB, NM

0060853

1. Report No. NASA CR-1672 call no.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Nonlinear Actuator Disk Theory and Flow Field Calculations, Including Nonuniform Loading		5. Report Date ~~September 1970~~ 6. Performing Organization Code
7. Author(s) Michael D. Greenberg and Stephen R. Powers		8. Performing Organization Report No.
9. Performing Organization Name and Address m.e Sage Action, Inc. ~~P. O. Box 416~~ ~~Ithaca, New York 14850~~		10. Work Unit No. SAIN — 721-01-10-04 add e 11. Contract or Grant No. NAS1-8173 omit
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		13. Type of Report and Period Covered ~~Contractor Report~~ 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

The axisymmetric flow induced by an actuator disk with prescribed nonuniform circulation distribution is considered. Coupled, nonlinear, singular integral equations governing the wake geometry and vortex density are developed from the force-free condition and discussed from physical and mathematical points of view. An iterative solution based partly on the method of successive approximations and partly on the Newton-Raphson method is put forward, together with convergent numerical results for illustrative cases, both static and nonstatic. Corresponding detailed flow field calculations are included.

17. Key Words (Suggested by Author(s)) Rotor Propeller Static Thrust Nonuniform Loading		18. Distribution Statement Unclassified - Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 82	22. Price* $3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

NONLINEAR ACTUATOR DISK THEORY AND FLOW FIELD CALCULATIONS, INCLUDING NONUNIFORM LOADING

By Michael D. Greenberg* and Stephen R. Powers
Sage Action, Inc.

SUMMARY

The axisymmetric flow induced by an actuator disk with prescribed nonuniform circulation distribution is considered. Coupled, nonlinear, singular integral equations governing the wake geometry and vortex density are developed from the force-free condition and discussed from physical and mathematical points of view. An iterative solution based partly on the method of successive approximations and partly on the Newton-Raphson method is put forward, together with convergent numerical results for illustrative cases, both static and nonstatic. Corresponding detailed flow field calculations are included.

INTRODUCTION

The increasing importance of "nonclassical" propeller theory over the past decade has led to the development of a substantial research effort, encompassing sophisticated mathematical models, performance tests and flow visualization studies, e.g. see Ref. 1. At the recent Third CAL/AVLABS Symposium, a comprehensive bibliography to earlier work was given in a paper by J. C. Erickson, Jr. (Ref. 2).

Most of the theoretical effort to date by others has been directed toward the development of finite-bladed models. In contrast, we have concentrated on the case of infinite blade number, that is, the actuator disk model. This model, of course, suffers from a number of inherent physical limitations. Two of these limitations will become clearer later on. One concerns the possibility of reverse flow at the blade tip and the other, the relationship of the resultant flow field to the mean or zeroth harmonic of the finite-bladed flow field.

---

* Now Assistant Professor, University Of Delaware, Dept. of Mechanical and Aerospace Engineering, and Staff Consultant to SAI.

On the other hand, the actuator disk model enjoys a substantial advantage in its relative simplicity. This greatly facilitates the derivation of both qualitative and quantitative information about the flow field, particularly the flow field external to the slipstream. As a result, it should prove to be a basic building block of understanding in the static and low speed regime, somewhat the same as it has for light loading in forward flight.

Linear theory for the actuator disk model has been established for some time, e.g. see Ref. 3. However, the first important paper on the *nonlinear* theory did not appear until 1962 by T. Y. Wu. In Ref. 4, he formulated the boundary value problem very elegantly and proposed an iterative scheme for its solution. As far as we know, this solution has not been completed.

Further work has been done by others along the lines of Wu's formulation. One analysis not reviewed in Ref. 2 is the analysis by B. W. Cox (Ref. 5) who used discrete vortex rings. Rather than concentrate on the details for static and low speed operation explicitly, he turned his attention to approximations for a number of other complications, such as a nonuniform free stream and wind tunnel wall interference.

In our first investigation of a nonlinear actuator disk theory (Ref. 6), we confined our efforts to the simplest case of a prescribed *uniform* circulation distribution. Taking an approach somewhat different than Wu's, we developed an equivalent vortex model. With this model, converged solutions were obtained for operation anywhere between the static condition and cruise.

The present study is, principally, an extension of the analysis of Ref. 6 to the more practical case of a *nonuniform* circulation distribution, which is approximated by a piecewise constant distribution. At the same time, though, a new and more physical derivation of the governing equations has been derived and significant improvements have been incorporated into the numerical analysis.

The authors would like to express their thanks to Dr. D. E. Ordway of SAI for his encouragement and helpful advice throughout the course of this work.

PRINCIPAL NOMENCLATURE

a_{kj}	coefficients in representation of vortex density for k^{th} vortex tube
b_{kj}	coefficients in representation of the shape of the kth vortex tube
A, B	coefficients in G_T, see Eqs. (14) and (15)
c_j	exponents in matching functions g_{kj}
C_T	thrust coefficient referenced to $\tilde{\rho}(\Omega R)^2(\pi R^2)$
D	slipstream region
f_j	matching functions for vortex tube shapes, see Eqs. (30) and (32)
F_k	twice the right hand side of Eq. (24)
g_{kj}	matching functions for vortex density of the kth vortex tube, see Eqs. (31) and (33)
G	Green's function for governing partial differential equation on Ψ
G_T	$\partial G/\partial T$
$G_{T_{\nu k}}$	shorthand for $G_T(\xi, t_\nu; x, T_k)$
$\underline{i}_r$, $\underline{i}_x$, ...	unit vectors in the r, x, ... directions, respectively
j	matching function index, e.g. see Eqs. (30) - (33)
k	index to distinguish the K different vortex tubes
K	total number of vortex tubes or, equivalently, the total number of steps in the piecewise constant circulation distribution
M	number of axial stations used in the stream function interpolation
N, N_K	number of collocation points used in solving for the vortex density on the Kth vortex tube
N', N_k	number of collocation points used in solving for the vortex density on the vortex tubes k = 1 through (K-1)
$\mathcal{N}$	number of blades
q	fluid velocity, $(u^2 + v^2 + w^2)^{1/2}$
$Q_{\pm 1/2}$	Legendre functions of second kind and degree $\pm 1/2$
R	propeller radius
R_k	radial location of the kth step in circulation distribution
s	meridional coordinate, see Fig. 3b
S	slipstream surface, see Fig. 3a

t, T	radius of Kth vortex tube, with arguments ξ and x, respectively
t_k, T_k	radius of kth vortex tube, with arguments ξ and x, respectively
u,v,w	x,r,θ velocity components
u',v',w'	x,r,θ perturbational velocity components
u_k	axial velocity on the kth vortex tube, see Eq. (25)
U	free-stream speed
x,r,θ	cylindrical coordinate system, see Fig. 1
α_ν, β_ν	axial stations used in solving for the vortex densities
γ, γ_s	vortex density on S for constant $\Gamma(r)$; circulation per unit x-length and s-length, respectively
γ_k, γ_{sk}	vortex density on k^{th} vortex tube; circulation per unit x-length and s-length, respectively
$\Gamma(r)$	circulation distribution
Γ_k	piecewise constant circulation distribution
δ_ν	axial stations used in stream function interpolation
ζ	meridional velocity, $(u^2+v^2)^{1/2}$
λ	advance ratio, $U/\Omega R$
ν	dummy index for k in t_k and γ_k
ξ,ρ	dummy x,r variables, respectively
$\tilde{\rho}$	fluid mass density
Ψ	stream function
$\underline{\omega}$	fluid vorticity vector with components $\omega_s,\omega_n,\omega_\theta$ in s,n,θ directions, respectively
$\tilde{\omega}$, $\tilde{\omega}_1$, $\tilde{\omega}_2$	Legendre function arguments, see Eq. (6) and pp. 10 and 54
Ω	propeller rotational velocity, radians per unit time
$(\ )_x$	subscripted variable denotes partial differentiation with respect to that variable
$(\ )_\infty$	asymptotic value at $x = \infty$
$(\ )^{(n)}$	n^{th} iterate
$\sqrt{h}$	shorthand for $[1+(dT/dx)^2]^{1/2}$, i.e. ds/dx

NOTE: Prior to Eq. (23), all quantities are in *dimensional* form. Starting with Eq. (23), they are all *nondimensionalized* as follows: lengths with respect to R; velocities, γ and γ_s with

respect to ΩR; Γ with respect to ΩR^2; and Ψ with respect to ΩR^3. *However, for notational simplicity we omit any explicit reminder of nondimensionalization, such as primes or asterisks.*

THEORETICAL DEVELOPMENT

Problem Statement And Formulation. Let us consider a propeller of blade radius R, operating with angular velocity Ω relative to a uniform free stream $U \geq 0$. With the blade circulation distribution prescribed, our objective is the calculation of the induced flow field.

We consider the blade number to be infinite, the so-called "actuator disk" model, and view the steady axisymmetric flow from a Newtonian x,r,θ coordinate system; see Fig. 1 in which we have sketched one of the infinitely many blades.

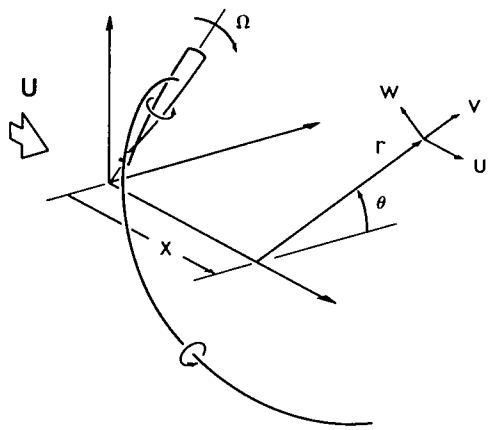

Figure 1. Coordinates And Geometry

The flow field, assumed to be inviscid and incompressible, is defined by the x,r,θ velocity components u,v,w respectively, or equivalently, by w and a stream function Ψ, such that

$$u = U + u' = \Psi_r / r \tag{1}$$

$$v = v' = - \Psi_x / r \tag{2}$$

where the primed terms are perturbational quantities and the subscripts indicate partial differentiation.

It has been shown by Wu (Ref. 4) that Ψ must satisfy the nonlinear partial differential equation

$$\Psi_{rr} - \Psi_r / r + \Psi_{xx} = - (\Omega r^2 + wr)\ d(wr)/d\Psi \tag{3}$$

An alternative derivation to Eq. (3) is based on the calculation of the circulation about an elemental meridional area dxdr in two different ways. That is, on the one hand, it may be computed from Stokes' Theorem as the θ component of vorticity, $(v_x - u_r)$, times the area dxdr or, on the other hand, as the line integral of "$\underline{q} \cdot d\underline{r}$" around the circumference of the element. Equating these two results produces Eq. (3). The details are given in Appendix A.1 .

The quantities wr and $d(wr)/d\Psi$ on the right hand side cause considerable complication since the functional dependence $wr = f(\Psi)$ is not known a priori. (We *do* know wr for $x = 0^+$ and $0 < r < R$ since the circulation distribution $\Gamma(r) = - 2\pi wr$ is prescribed there, but we do *not* know wr as a function of Ψ throughout the field.) Whereas Kelvin's Theorem implies that $f(\Psi) = 0$ outside the slipstream region D, we do not know the shape of D at the outset.

Due to these complications, an analytical solution of the resulting nonlinear free-boundary problem seems to be out of the question and we choose instead to pursue an iterative approach. Toward this end, it is helpful to first recast Eq. (3) in the form of an *integral* equation, which can be expected to be more amenable to numerical solution. This has already been done (Refs. 4 and 5) and the resulting integral equation can be expressed in the form,

$$\Psi(x,r) = Ur^2/2 + \iint_D G(\xi,\rho;x,r)\ (\Omega\rho + w)\ d(w\rho)/d\Psi\ d\rho d\xi \tag{4}$$

The Green's function,

$$G(\xi,\rho;x,r) = \sqrt{r\rho}\ Q_{\frac{1}{2}}(\tilde{\omega})/2\pi \tag{5}$$

may be identified as the stream function at a field point (x,r,θ) due to a ring vortex of unit strength, oriented as shown in Fig. 2, where $Q_{\frac{1}{2}}(\tilde{\omega})$ is a Legendre function of second kind and degree ½, with argument

$$\tilde{\omega} = 1 + [(\xi-x)^2 + (\rho-r)^2] / 2\rho r \tag{6}$$

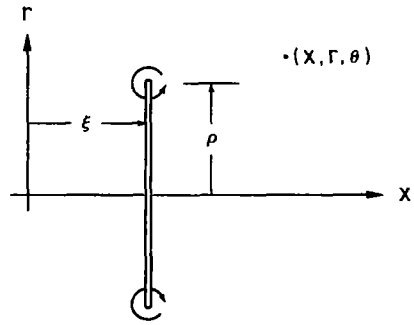

Figure 2. Interpretation Of The Green's Function

Recall that we mentioned that the flow field is completely determined by the stream function $\Psi(x,r)$ *and* the tangential velocity field $w(x,r)$. Once Ψ is calculated from Eq. (4), w can be found throughout D by application of Kelvin's Theorem. The hard part, of course, is the calculation of Ψ. This will be the subject of the following sections.

Vortex Model. Whereas Wu deals with Eq. (4) directly, we elected to work with an equivalent vortex model and limit our attention to piecewise constant blade circulation distributions. That is, we prescribe constants $R_1, \ldots, R_K$ and $\Gamma_1, \ldots, \Gamma_K$ such that $\Gamma(r) = \Gamma_k$ over $R_{k-1} < r < R_k$, where $R_0 = 0$ and $R_K = R$. Consequently, $d(wr)/d\Psi$ is zero everywhere in D, except on the streamtubes, or vortex tubes, which pass through $x = 0$, $R = R_1, \ldots, R_K$. On these streamtubes, it has a delta function type behavior. The ρ integration thus reduces to a finite summation over the discrete vortex tubes and Eq. (4) may be re-expressed formally as

$$\Psi(x,r) = Ur^2/2 + \sum_{k=1}^{K} \int_0^{\infty} G(\xi,t_k;x,r)\ \gamma_k\ d\xi \tag{7}$$

where $t_k = t_k(\xi)$ and $\gamma_k = \gamma_k(\xi)$ are the radius and circulation per unit ξ-length, respectively, of the kth vortex tube. Our objective, then, is the calculation of the unknown "shapes" $t_k(\xi)$ and "vortex densities" $\gamma_k(\xi)$.

At this point, we would like to relate the local disk loading to the circulation distribution. If we refer to the analysis of Ref. 7, we see that

the radial distribution of circulation will be essentially proportional to the radial distribution of annulus or local disk loading. The local disk loading, of course, varies as the blade loading divided by r.

Force-Free Condition. Whereas the boundary value problem on $\Psi(x,r)$ is completely contained in Eq. (4), the unknown $t_k(\xi)$'s and $\gamma_k(\xi)$'s are *not* determined by Eq. (7) because we have, in going from Eq. (4) to Eq. (7), taken certain terms defined in terms of Ψ and simply called them $\gamma_k(\xi)$. This, in effect, introduced additional unknowns.

In order to obtain a sufficient number of equations, it is necessary to impose the physical condition that the trailing vorticity drifts force-free. More specifically, the trailing vortices must be *aligned with streamlines* if the Kutta-Joukowski forces on them are to be precluded. It can, in fact, be verified that this is just the information that was "lost" above.

Since the extension to the piecewise constant case can be accomplished easily, let us first consider the case where $\Gamma(r)$ is a constant, say Γ. The vortex system then is as sketched in Fig. 3a. It consists of an infinite

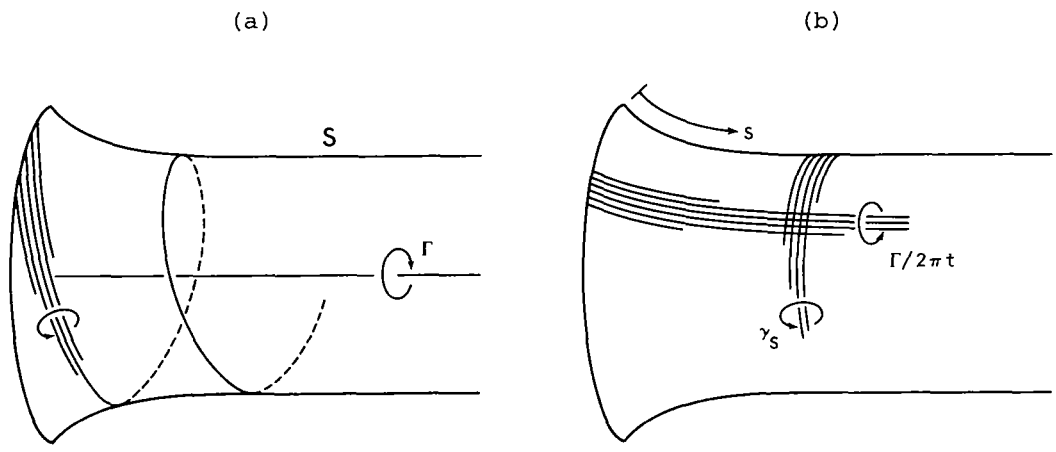

Figure 3. Vortex System For Constant $\Gamma(r)$

number of radial vortex lines which represent the blades, not shown here, together with an equal number of distorted helical vortices trailing from the blade tips and a concentrated "hub" vortex of strength Γ on the positive x axis. All of these vortices are to be force-free except, of course, the

radial or bound blade vortices.

As shown in Fig. 3b, the trailing vorticity may be decomposed into an equivalent orthogonal system consisting of ring vortices plus vortices lying along the intersection of the slipstream surface S and the meridional planes of constant θ. Let the ring vortices be of strength γ_s per unit s-length or, alternatively, γ per unit x-length. From conservation of vorticity we see that the strength of the meridional vorticity is equal to Γ divided by the local circumference, or $\Gamma/2\pi t$ per unit tangential length as indicated.

Alignment of the trailing vorticity with the streamlines requires, first, that the velocity component *normal* to S be zero everywhere on this surface, or*

$$\Psi(x,T) = UT^2/2 + \int_0^\infty G(\xi,t;x,T)\ \gamma\ d\xi = \Psi(0,R) \tag{8}$$

and, second, that

$$(\Gamma/2\pi T)\ /\ \gamma_s = \zeta\ /\ (\Omega T + w) \tag{9}$$

on $r = T(x)$, where ζ is the meridional velocity component $(u^2 + v^2)^{1/2}$. Since $\gamma_s\zeta = \gamma u$ and $w(T) = [w(T+0) + w(T-0)]/2 = (0 - \Gamma/2\pi T)/2$, the condition of Eq. (9) becomes

$$\gamma u = \Omega\Gamma/2\pi - \Gamma^2/8\pi^2T^2 \tag{10}$$

on $r = T(x)$. Thus, if we substitute $u = \Psi_r / r$, we have finally

$$\gamma\left[U + \frac{1}{T}\int_0^\infty G_T(\xi,t;x,T)\ \gamma\ d\xi\right] = \Omega\Gamma/2\pi - \Gamma^2/8\pi^2T^2 \tag{11}$$

on $r = T(x)$, where the integral is to be evaluated in the Cauchy principal value sense.

We now see that the coupled, singular, nonlinear integral equations of Eqs. (8) and (11) suffice to determine the two unknowns, $T(x)$ and $\gamma(x)$. The kernels are as follows. For Eq. (8), we have from Eq. (5)

---

* It will be convenient to express the slipstream radius as t or T depending on whether the argument is the integration variable ξ or the field point x, respectively.

$$G = \sqrt{Tt}\; Q_{\frac{1}{2}}(\tilde{\omega}_1)/2\pi$$

$$= O(\ell n|\xi - x|) \quad , \quad \xi \to x$$

$$= O(\xi^{-3}) \quad , \quad \xi \to \infty \tag{12}$$

where $\tilde{\omega}_1$ is identical to $\tilde{\omega}$, with ρ and r replaced by t and T, respectively. Using the relation,

$$dQ_{\frac{1}{2}}(z)/dz = [zQ_{\frac{1}{2}}(z) - Q_{-\frac{1}{2}}(z)] / 2(z^2 - 1) \tag{13}$$

we find for Eq. (11),

$$G_T = [A\,Q_{\frac{1}{2}}(\tilde{\omega}_1) + B\,Q_{-\frac{1}{2}}(\tilde{\omega}_1)] / (\tilde{\omega}_1^2 - 1)$$

$$= O(\xi - x)^{-1} \quad , \quad \xi \to x$$

$$= O(\xi^{-3}) \quad , \quad \xi \to \infty \tag{14}$$

where A and B are given by,

$$A = [T^2 - t^2 + (\xi - x)^2] / 8\pi\sqrt{Tt^3}$$

$$B = [t^2 - T^2 + (\xi - x)^2] / 8\pi\sqrt{tT^3} \tag{15}$$

As we said, satisfaction of Eq. (8) precludes any flow *normal* to S, and hence any Kutta-Joukowski forces directed *tangent* to that surface. Alignment of the vorticity and velocity vectors *on* S, guaranteed by Eq. (11), further precludes any Kutta-Joukowski forces directed *normal* to that surface. Thus, Eq. (11) must be equivalent to the statement that the pressure difference across S is zero and this equation could have been derived by application of Bernoulli's Equation. A derivation along these lines can be found in Ref. 6.

Notice that insofar as the two governing equations are concerned, the effect of swirl, i.e., the induced tangential velocity, is limited to the presence of the second term in the right hand side of Eq. (10). Since, for Γ's and Ω's within the range of general interest, this term turns out to be negligible compared to $\Omega\Gamma/2\pi$, the neglect of swirl by other investigators, e.g. Refs. 5 and 8, is not unreasonable. We choose to retain it, however,

because its presence in no way complicates our solution and, more importantly, we suspect that it may not be negligible for the innermost vortex tubes in the case of *non*-constant circulation. We will clarify this point when we discuss our numerical results.

Relation Between Actuator Disk Model And Zeroth Harmonic Flow. Before we go ahead, let us pause to consider a question of both fundamental and practical interest. Namely, *is the actuator disk flow equal to the zeroth circumferential harmonic of the flow field induced by a finite N-bladed propeller having the same* $\Gamma(r)$ *distribution*, as found in linearized theory (Ref. 3)?

For simplicity, we consider the case of constant $\Gamma(r)$ again and visualize the axisymmetric surface S which contains the N trailing tip vortices. As above, the force-free condition requires the alignment of the vorticity vector $\underline{\omega} = \nabla \times \underline{q}$ and the velocity $\underline{q}$ relative to that vorticity, hence relative to a blade-fixed coordinate system, over the surface S to which the tip-trailing vorticity is confined. In other words,

$$\underline{\omega} \times \underline{q} = \begin{vmatrix} \underline{i}_s & \underline{i}_n & \underline{i}_\theta \\ \omega_s & \omega_n & \omega_\theta \\ \zeta & \underline{q}\cdot\underline{i}_n & \Omega T+w \end{vmatrix} = 0 \tag{16}$$

on S, where the unit vectors are orthonormal, with $\underline{i}_n$ normal to S and $\underline{i}_s$ and $\underline{i}_\theta$ tangent to S in the s and θ directions respectively. Noting that $\omega_n = 0$, we have the three scalar equations

$$\omega_\theta(\underline{q}\cdot\underline{i}_n) = 0 \tag{17}$$

$$\omega_s(\underline{q}\cdot\underline{i}_n) = 0 \tag{18}$$

$$\omega_s(\Omega T+w) - \omega_\theta\zeta = 0 \tag{19}$$

As $N \to \infty$, $\omega_\theta \to \gamma_s$, $\omega_s \to \Gamma/2\pi T$ and $w \to -\Gamma/4\pi T$, so that both Eqs. (17) and (18) imply that $\underline{q}\cdot\underline{i}_n = 0$ on S. This coincides with the previous condition of Eq. (8), and Eq. (19) reduces to the condition of Eq. (9), thus recovering our actuator disk equations.

Turning now to finite N, we only need to discuss any one of the three equations, Eqs. (17) - (19), for example, Eq. (17). Now, ω_θ will be zero for all θ's except at the N circumferential trailing vortex locations, where it will have a delta function character such that its zeroth, N^{th}, $2N^{th}$, ..., harmonics will be nonzero. The idea, therefore, is to expand ω_θ and $\underline{q}\cdot\underline{i}_n$ *into these harmonics, multiply, and equate the resulting coefficients of the* different harmonics to zero. If the resulting zeroth harmonic is simply the zeroth harmonic of ω_θ times the zeroth harmonic of $\underline{q}\cdot\underline{i}_n$, then equating this to zero would imply that the zeroth harmonic of $\underline{q}\cdot\underline{i}_n$ is zero — in agreement with our actuator disk flow. However, additional contributions to the zeroth harmonic can arise through the product of higher harmonics; for example, we have $\cos N\theta \cos N\theta = (\frac{1}{2} + \frac{1}{2}\cos 2N\theta)$. Since there is no reason why these contributions should exactly cancel each other, it appears that the zeroth harmonic of $\underline{q}\cdot\underline{i}_n$ may *not* be zero. In that case the actuator disk flow would *not* equal the zeroth harmonic of the flow field induced by a finite-bladed propeller.

For the special case of *light loading*, i.e., the classical linearized theory, though, the perturbational velocities can be neglected and the trailing vortices are undeformed so that $\underline{q}\cdot\underline{i}_n = U\underline{i}_x\cdot\underline{i}_n = 0$, and Eqs. (17) - (19) become

$$0 = 0 \tag{20}$$

$$0 = 0 \tag{21}$$

$$\omega_s\Omega T - \omega_\theta U = 0 \tag{22}$$

Since ΩT and U are independent of θ, we have no products of higher harmonics and the identity of the actuator disk with the zeroth harmonic is again established for the linearized case, cf. Ref. 3.

These results imply that caution must be used in trying to apply actuator disk results for cases with both low blade number, $N < 6$ or so, *and* heavy loading. This is especially true with respect to the inflow over the outer portions of the blades.

Extension To The Case Of Piecewise Constant Circulation. First, let us put our equations in nondimensional form, referring lengths to R; velocities, γ and γ_s to ΩR; Γ to ΩR^2; and Ψ to ΩR^3. *For convenience, we will omit any explicit notation in the remainder of the report to distinguish these quantities from their dimensional form.*

Now, extension of Eqs. (8) and (10) for the case of piecewise constant circulation is quite straightforward and results in the equations

$$\Psi(0,R_k) = \lambda T_k^2/2 + \sum_{\nu=1}^{K} \int_0^\infty G(\xi,t_\nu;x,T_k)\ \gamma_\nu\ d\xi \tag{23}$$

$$\gamma_k u_k = (\Gamma_k - \Gamma_{k+1})/2\pi - (\Gamma_k^2 - \Gamma_{k+1}^2)/8\pi^2 T_k^2 \tag{24}$$

to be satisfied on each vortex tube $r = T_k(x)$. The advance ratio λ is defined as $U/\Omega R$ and

$$u_k = \lambda + \frac{1}{T_k} \sum_{\nu=1}^{K} \int_0^\infty G_T(\xi,t_\nu;x,T_k)\ \gamma_\nu\ d\xi \tag{25}$$

It is understood that $\Gamma_{K+1} = 0$.

Eqs. (23) and (24) constitute the final set of 2K coupled, nonlinear, singular integral equations in the 2K unknowns T_k and γ_k, for k = 1 through K.

Asymptotic Behavior Of The Unknowns. Let us examine the equations at $x = 0$ and ∞, starting with $x = \infty$.

As $x \to \infty$, it is known that $T_k(x)$ approaches a constant, say $T_{k\infty}$, and $\gamma(x)$, say $\gamma_{k\infty}$. With these quantities constant at $x = \infty$, the integrals in Eqs. (23) - (25) can be evaluated analytically. Instead of pursuing the details of these integrations, we use the known result (e.g., Ref. 3) that an infinite solenoid of constant radius and constant vortex density, $T_{k\infty}$ and $\gamma_{k\infty}$ in our case, induces a velocity field given by

$$\begin{aligned}(u',v',w') &= (0,0,0) &&, \quad r > T_{k\infty} \\ &= (\gamma_{k\infty}/2,0,0) &&, \quad r = T_{k\infty} \\ &= (\gamma_{k\infty},0,0) &&, \quad r < T_{k\infty}\end{aligned} \tag{26}$$

Since $2\pi\Psi(\xi,\rho)$ is the dimensionless volume flow through the disk $r < \rho$ at $x = \xi$, we then see by virtue of Eq. (26) that Eq. (23) *must* reduce to

$$2\Psi(0,R_k) = \lambda T_{k\infty}^2 + T_{k\infty}^2 \sum_{\nu=k}^{K} \gamma_{\nu\infty} + \sum_{\nu=1}^{k-1} T_{\nu\infty}^2 \gamma_{\nu\infty} \tag{27}$$

as $x \to \infty$.

In order to look at Eq. (24) as $x \to \infty$, we have with the help of Eq. (26) that $u_k \sim (\lambda + \frac{1}{2}\gamma_{k\infty} + \gamma_{k+1,\infty} + \ldots + \gamma_{K\infty})$. Calling the right hand side of Eq. (24), $\frac{1}{2}F_k$, we can write the limiting form of this equation therefore as

$$\begin{aligned}
k = K &: \quad \gamma_{K\infty}(2\lambda + \gamma_{K\infty}) = F_{K\infty} \\
k = K-1 &: \quad \gamma_{K-1,\infty}(2\lambda + \gamma_{K-1,\infty} + 2\gamma_{K\infty}) = F_{K-1,\infty} \\
&\vdots \\
k = 1 &: \quad \gamma_{1\infty}(2\lambda + \gamma_{1\infty} + 2\gamma_{2\infty} + \ldots + 2\gamma_{K\infty}) = F_{1\infty}
\end{aligned} \tag{28}$$

where $F_{k\infty}$ is simply F_k at $x = \infty$, that is, with T_k replaced by $T_{k\infty}$. These equations are K simultaneous algebraic equations in the asymptotic values $\gamma_{1\infty}$ through $\gamma_{K\infty}$. Their form is such that they can be solved successively for $\gamma_{K\infty}$, $\gamma_{K-1,\infty}$, ..., $\gamma_{1\infty}$, yielding

$$\gamma_{k\infty} = -\Lambda + \sqrt{\Lambda^2 + F_{k\infty}} \quad , \quad \Lambda \equiv \lambda + \sum_{\nu=k+1}^{K} \gamma_{\nu\infty} \tag{29}$$

for $k = K, K-1, \ldots, 1$ in turn. It is understood that $\Lambda = \lambda$ for $k = K$.

Of these various results, Eq. (27) will be used to put $\Psi(0,R_k)$ in Eq. (23) in terms of the unknown T_k's and γ_k's, and Eq. (29) will be used in constructing a suitable analytic form for our solution.

It is important to note that the asymptotic relations of Eqs. (27) and (29) constitute 2K equations in 3K unknowns, the $\Psi(0,R_k)$'s, $T_{k\infty}$'s and $\gamma_{k\infty}$'s. Thus, *the final slipstream contraction, for example, cannot be computed immediately in terms of the operating conditions simply by investigation of the asymptotic behavior, but must await the complete solution of the governing integral equations.*

Consider now the behavior of the vortex densities as $x \to 0^+$. For $\lambda = 0$ the flow in each meridional plane makes a full turn around the slipstream lip. Consequently, it is clear that a classical square root singularity is present in the outermost vortex density, so that $\gamma_K(x) = O(x^{-1/2})$ as $x \to 0^+$. This singularity should, we believe, persist even in the *non*static case, $\lambda > 0$, and vanish only in the light loading limit when the induced flow is negligible compared to the free stream and the $\gamma_k(x)$'s and $T_k(x)$'s are all constant. On the other hand, we anticipate that the inner vortex tubes will not be singular, so that $\gamma_k(x) = O(1)$ as $x \to 0^+$ for all $k < K$.

Assumed Form Of The Unknowns. It will be convenient to assume the following approximate form for the unknowns,

$$T_k(x) = R_k + \sum_{j=1}^{M} b_{kj}\, f_j(x) \tag{30}$$

$$\begin{aligned} \gamma_k(x) &= [1 + (dT_k/dx)^2]^{1/2}\, \gamma_{sk}(x) \\ &= "(\sqrt{\pi})_k"\, \gamma_{k\infty} \left\{ 1 + \sum_{j=1}^{N_k} a_{kj}\, g_{kj}(x) \right\} \end{aligned} \tag{31}$$

where the f_j's and g_{kj}'s are "matching functions". All of these functions are chosen to be regular, except for g_{K1}, which provides the necessary square root singularity for γ_K at $x = 0$. In selecting suitable matching functions, we require that $f_j(0) = 0$ so that $T_k(0) = R_k$. We also require that all of the g_{kj}'s $\to 0$ as $x \to \infty$. With $dT_k/dx \sim 0$ this ensures that $\gamma_k(x) \sim \gamma_{k\infty}$.

The reason for the k subscript on N is that we would like to be able to do a particularly good job in calculating γ_K, which is generally quite a bit larger than the other γ_k's, and singular as well. In practice, we have found it efficient to use $N_k \approx 6$ for all k's $< K$ and $N_K \approx 9$. Since we will always use $N_1 = N_2 = \ldots = N_{K-1}$, let us simply denote these values by a single name, say N', and N_K simply by N.

By trial and error over a number of calculations, we have decided upon the following set of matching functions,

$$f_j(x) = e^{-jx} - 1$$

$$g_{kj}(x) = x^{-\frac{1}{2}} e^{-3x} \quad , \quad k = K \quad j = 1$$

$$= x^{c_j} e^{-3x} \quad , \quad k = K \quad j \geq 2$$

$$= e^{-jx} \quad , \quad k < K \tag{33}$$

with the c_j's positive constants ranging from about 0.2 to about 4.

General Outline Of Iterative Solution. We will first outline our iterative solution in general terms. Starting with zeroth iterates which are constant, namely,

$$T_k^{(0)}(x) = R_k \tag{34}$$

$$\gamma_k^{(0)}(x) = \gamma_{k\infty}^{(0)} \tag{35}$$

we compute improved vortex tube shapes $T_k^{(1)}$ based upon Eq. (23). With $T_k = T_k^{(1)}$ we then compute improved vortex densities $\gamma_k^{(1)}$ from Eq. (24). With $\gamma_k = \gamma_k^{(1)}$ we, in turn, compute $T_k^{(2)}$'s, and so on until suitable convergence is attained. Our notation is to be interpreted in the obvious way. For example, $\gamma_{k\infty}^{(0)}$ is given by Eq. (29) with $T_{k\infty}^{(0)}$ set equal to R_k in $F_{k\infty}^{(0)}$.

In Ref. 6 the improved shapes and vortex densities were obtained by applying a Newton-Raphson technique to Eqs. (23) and (24), respectively. For the present case, where $K > 1$, we found this scheme to be generally inappropriate for the vortex tube shape calculations since closely spaced tubes usually crossed each other, producing a divergent situation. Instead of a Newton-Raphson technique, we now proceed along different lines.

Vortex Tube Shape Calculation. Regarding the $T_k^{(n)}$'s and $\gamma_k^{(n)}$'s as known, we determine the $T_k^{(n+1)}$'s as follows. We first compute $\Psi^{(n)}(0,R_k)$ for $k = 1, \ldots, K$ and then seek these stream function values at each of M selected axial stations, $x = \delta_1, \ldots, \delta_M$. To accomplish this, at a given station δ_ν we calculate $\Psi^{(n)}$ at the radial points $T_1^{(n)}(\delta_\nu)/2$, $T_1^{(n)}(\delta_\nu)$, $[T_1^{(n)}(\delta_\nu) + T_2^{(n)}(\delta_\nu)]/2$, $\ldots$, $[T_{K-1}^{(n)}(\delta_\nu) + T_K^{(n)}(\delta_\nu)]/2$ and $T_K^{(n)}(\delta_\nu)$, and then use three point radial interpolation to find the radius at which each of the values $\Psi^{(n)}(0,R_k)$ result. With these points in hand, say $T_{k\nu}^{(n+1)}$, we get the $b_{kj}^{(n+1)}$'s by "fitting". That is, recalling Eq. (30), we have

$$T_{k\nu}^{(n+1)} - R_k = \sum_{j=1}^{M} b_{kj}^{(n+1)} f_j(\delta_\nu) \tag{36}$$

where $\nu = 1, \ldots, M$ for each fixed k. Inversion of these K $M \times M$ matrix equations yields the desired $b_{kj}^{(n+1)}$'s.

It may happen that one or more of the $\Psi^{(n)}$'s *cannot be found* in the radial search at a given δ_ν station. Consider the case where $K = 1$. If $\Psi^{(n)}(0,1) > \Psi^{(n)}[\delta_\nu, T_1^{(n)}(\delta_\nu)]$ and λ is close enough to zero or the static case, we will not be able to find $\Psi^{(n)}(0,1)$ in our radial search at $x = \delta_\nu$ because the stream function drops off as we go out radially beyond $T_1^{(n)}(\delta_\nu)$. At such a location we should choose $T_1^{(n+1)}(\delta_\nu)$ sufficiently larger than $T_1^{(n)}(\delta_\nu)$ so as to gain the required additional volume flow. In particular, we put

$$2\pi \left\{ \Psi^{(n)}(0,1) - \Psi^{(n)}[\delta_\nu, T_1^{(n)}(\delta_\nu)] \right\} =$$

$$\pi \left\{ [T_1^{(n+1)}(\delta_\nu)]^2 - [T_1^{(n)}(\delta_\nu)]^2 \right\} u \tag{37}$$

Now, $u \approx [\lambda + \gamma_{1\infty}^{(n)}]$ for δ_ν large, whereas $u \approx [\lambda + \gamma_{1\infty}^{(n)}/2]$ for δ_ν small. Therefore, let us use the approximation

$$u \approx \lambda + \left(\frac{\delta_\nu + 0.15}{\delta_\nu + 0.30} \right) \gamma_1^{(n)}(\delta_\nu) \tag{38}$$

and Eq. (37) can be solved for $T_1^{(n+1)}(\delta_\nu)$. Of course, any error induced by the approximation of Eq. (38) will tend to zero as the iteration converges.

As we have already said, for $K > 1$ it is conceivable that a number of the outer $\Psi^{(n)}$'s cannot be found. Adopting the same argument as described above for $K = 1$, we arrive at the following logic. If we cannot find $\Psi^{(n)}(0,R_k)$ at δ_ν, we set

$$T_k^{(n+1)}(\delta_\nu) = \sqrt{[T_k^{(n)}(\delta_\nu)]^2 + 2\{\Psi^{(n)}(0,R_k) - \Psi^{(n)}[\delta_\nu, T_k^{(n)}(\delta_\nu)]\} / u}$$

$$u \approx \lambda + \left(\frac{\delta_\nu + 0.15}{\delta_\nu + 0.30} \right) \left\{ \gamma_k^{(n)}(\delta_\nu) + \ldots + \gamma_K^{(n)}(\delta_\nu) \right\} \tag{39}$$

As our final comment on the shape calculation, we note that the radial three point interpolation interval should not be permitted to *cross* a vortex tube. For example, T_3, $(T_3 + T_4)/2$ and T_4 *would* be a suitable set of interpolation points whereas $(T_3 + T_4)/2$, T_4 and $(T_4 + T_5)/2$ cross the fourth vortex tube and therefore would *not* be suitable. The reason is that the stream function, considered as a function of radius, has abrupt changes in slope, or "kinks", when we cross a vortex tube, so that parabolic interpolation would not be very accurate.

Vortex Density Calculation. Take $k < K$. For the class of circulation distributions of practical interest, it turns out that the contribution to u_k in Eq. (24) by the kth vortex tube itself is small compared to the contribution by the Kth vortex tube. This implies that the dependence of u_k on γ_k is rather weak and suggests that a simple successive approximation scheme, namely,

$$\gamma_k^{(n+1)} = (F_k / 2u_k)^{(n)} \tag{40}$$

should suffice. The superscript (n) on the right hand side is somewhat misleading. Remember according to our overall scheme, we calculate the $T_k^{(n+1)}$'s and *then* the $\gamma_k^{(n+1)}$'s. As a result, when we reach Eq. (40), the $T_k^{(n+1)}$'s are already available and we might as well use them. Thus, the right hand side of Eq. (40) is computed with the $T_k^{(n+1)}$'s and the $\gamma_k^{(n)}$'s, and by means of Eqs. (31) and (33), this equation may be written in its final form,

$$\sum_{j=1}^{N'} a_{kj}^{(n+1)} \; e^{-jx} = \left\{ F_k / 2\gamma_{k\infty} (\sqrt{\hbar})_k \; u_k - 1 \right\}^{(n)} \tag{41}$$

where u_k is given by Eq. (25) and the superscript (n) on the right is interpreted as indicated above. For each $k = 1, 2, \ldots, (K-1)$ we demand satisfaction of Eq. (41) at the discrete points $x = \alpha_1, \ldots, \alpha_{N'}$. This produces $(K-1)$ $N' \times N'$ matrix equations which are then solved for the unknown $a_{kj}^{(n+1)}$'s.

For $k = K$, however, u_K is *not* weakly dependent on γ_K, so that a simple successive approximation scheme can be expected to be only very slowly convergent at best, or divergent at worst. We therefore choose to adapt the more powerful Newton-Raphson Method, as developed in Ref. 6. Briefly, we

re-express Eq. (24) as $u_K = F_K / 2\gamma_K$ and expand the nonlinear term $1/\gamma_K$ about the nth iterate,

$$1/\gamma_K^{(n+1)} \approx 1/\gamma_K^{(n)} - [1/\gamma_K^{(n)}]^2 [\gamma_K^{(n+1)} - \gamma_K^{(n)}] \tag{42}$$

Since u_K as well as the right hand side of Eq. (42) are *linear* in the unknown $\gamma_K^{(n+1)}$, the re-expressed form of Eq. (24) can be reduced to a system of algebraic equations in the unknown $a_{Kj}^{(n+1)}$'s. Omitting the details, we obtain

$$\sum_{j=1}^{N} \left(F_K T_K g_{Kj} (\sqrt{\hbar})_K / 2\gamma_K^2 + \int_0^\infty G_{T_{KK}} g_{Kj} (\sqrt{\hbar})_K \, d\xi \right)^{(n)} a_{Kj}^{(n+1)} =$$

$$\left(\left[\{2\gamma_K - \gamma_{K\infty} (\sqrt{\hbar})_K\} \{F_K / 2\gamma_K^2\} - \lambda \right] (T_K / \gamma_{K\infty}) - \right.$$

$$\left. \int_0^\infty G_{T_{KK}} (\sqrt{\hbar})_K \, d\xi - \frac{1}{\gamma_{K\infty}} \sum_{\nu=1}^{K-1} \int_0^\infty G_{T_{\nu K}} \gamma_\nu \, d\xi \right)^{(n)} \tag{43}$$

where $G_{T_{\nu k}}$ is a shorthand notation for $G_T(\xi, t_\nu; x, T_k)$. Again the superscript (n) is used, but the quantities involved are to be computed taking the $n+1^{th}$ shapes and n^{th} vortex densities. Demanding satisfaction of Eq. (43) at the discrete points $x = \beta_1, \ldots, \beta_N$ produces an $N \times N$ matrix equation for the unknown $a_{Kj}^{(n+1)}$'s.

We wish to emphasize that the approximation of Eq. (42) tends to an *equality* as the iteration converges, so that the essential nonlinear character of Eq. (24) is *not* compromised by the expansion of Eq. (42).

Finally, we note that the numerical calculation of the Legendre functions $Q_{\pm\frac{1}{2}}$, involved in G and G_T, is discussed in Appendix A.2 and the numerical integration scheme in Appendix A.3 . Descriptions and listings of the various computer codes that were developed for the general calculations are given in Appendix A.4

NUMERICAL RESULTS

Thrust Coefficient. In selecting circulation distributions for our numerical examples, we will take two "representative" distributions and scale their magnitude so that the resulting thrust coefficient C_T is within the range

of engineering interest. Defining C_T as the thrust divided by $\tilde{\rho}(\Omega R)^2(\pi R^2)$, with $\tilde{\rho}$ the fluid mass density, we have from the Kutta-Joukowski Law,

$$C_T = \frac{1}{\pi}\int_0^1 \left\{ r - \frac{\Gamma(r)}{4\pi r} \right\} \Gamma(r)\, dr \tag{44}$$

Representative Circulations. Let us consider the two representative distributions $\Gamma(r) = \Gamma$, a constant, and $\Gamma(r) = A\, r\sqrt{1-r}$ (Ref. 3), with the constants Γ and A chosen so that $C_T = 0.01$, say, which is a typical value for a hovering helicopter.

For Case 1, $\Gamma(r) = \Gamma$, the swirl term $\Gamma/4\pi r$ causes the integral to diverge. In reality, of course, a finite hub would preclude this divergence. In selecting a value for Γ, then, we simply neglect the swirl term in Eq. (44) and $C_T = 0.01 \approx \Gamma/2\pi$. Let us therefore take $\Gamma = 0.02\pi$.

For Case 2, $\Gamma(r) = A\, r\sqrt{1-r}$, the integral does exist and A = 0.21 results in the desired thrust coefficient $C_T = 0.01$. Next, see Fig. 4, we approximate

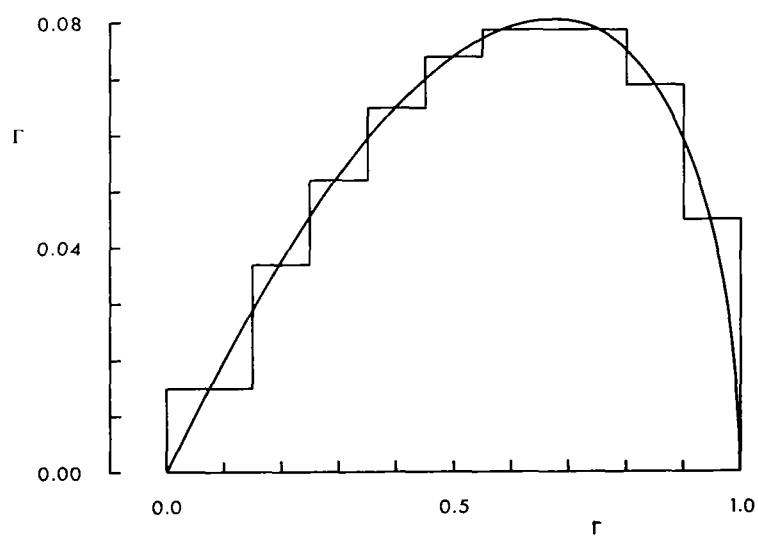

Figure 4. Representative Nonuniform Circulation And Its Piecewise Constant Approximation

$\Gamma(r)$ by a piecewise constant representation. For our purposes, an eight step representation seems to permit a suitably close fit, namely,

k	1	2	3	4	5	6	7	8	
R_k	0.15	0.25	0.35	0.45	0.55	0.80	0.90	1.00	(45)
Γ_k	0.015	0.037	0.052	0.065	0.074	0.079	0.069	0.045	

We do not claim that this *particular* representation provides a "best" fit in any special sense, but just that it appears to be entirely reasonable. Although there is a fair amount of latitude in the choice of the R_k's and Γ_k's, we must have $\Gamma_1 > 0$ for reasons of convergence that will be discussed later.

Case 1 Results And Discussion. Let us first look at the uniform circulation case where $K = 1$, $R_1 = 1.0$, $\Gamma_1 = 0.02\pi$ and $\lambda = 0$. To define our numerical solution, we chose $N = 9$; the β_ν's = 0.02, 0.05, 0.10, 0.18, 0.30, 0.50, 0.85, 1.40, 2.50; the c_j's = 0.18, 0.69, 1.20, 1.71, 2.22, 2.73, 3.24, 3.75; $M = 6$; the δ_ν's = 0.1, 0.3, 0.6, 1.0, 2.0, 5.0; and the $a_{1j}^{(0)}$'s, $b_{1j}^{(0)}$'s = 0, so that $T_1^{(0)}(x) = 1$ and $\gamma_1^{(0)}(x) = 0.14107$.

As seen from Fig. 5, the iteration is very nicely convergent, with $T_1(x)$ essentially converged after only two iterations, and $\gamma_1(x)$ after three. The NSWTH input variable, explained in Appendix A.3, was set equal to 3 so that the more accurate integration schemes were employed, beginning with the third iteration. The results of the fourth and final iteration are

$$a_{1j}^{(4)} = 0.14676,\ -2.70486,\ 21.20489,\ -98.70655,\ 272.84574,\ -428.19646,\ 374.74065,\ -170.60165,\ 31.97000$$

$$b_{1j}^{(4)} = -0.11037,\ 1.30353,\ -5.85472,\ 12.71052,\ -12.66851,\ 4.87670$$

for $j = 1$ through 9, and 1 through 6, respectively.

Representative streamtubes and meridional velocities have also been computed, using the results of the fourth iterate, and are presented in Fig. 6.

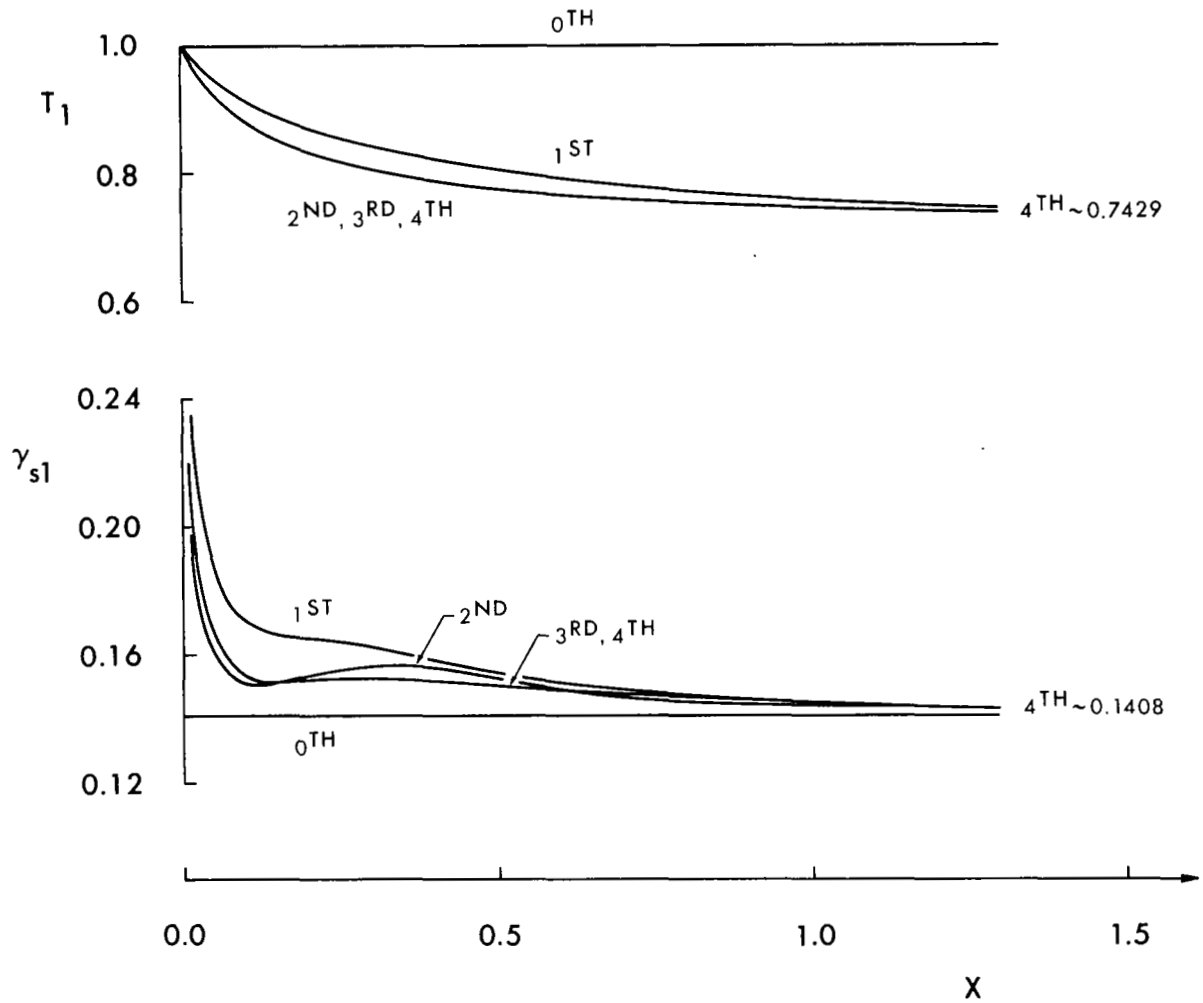

Figure 5. Successive Iterates For Case 1, $\lambda = 0$

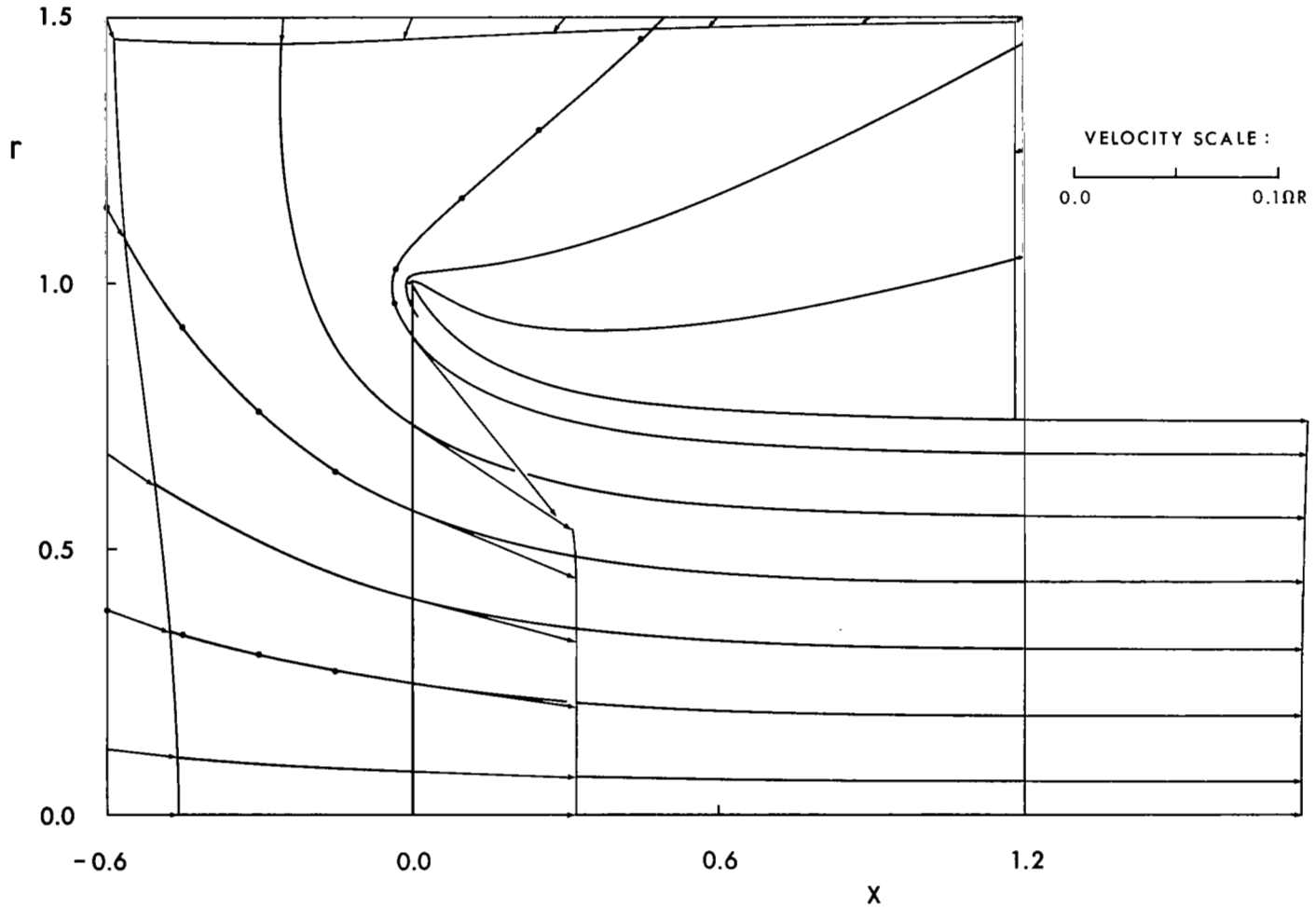

Figure 6. Streamtubes And Velocity Box For Case 1, $\lambda = 0$

To emphasize the appreciable difference between these results and those of the classical linearized theory, we have compared the respective streamtubes in Fig. 7. Actually, the term "linearized theory" does not make any sense for the static condition since there is no stream about which to linearize. But, in the linear theory it is known (Ref. 3) that the axial inflow $u'(0,r)$ is proportional to the circulation distribution $\Gamma(r)$. We can extend this idea down to the static condition and shall *define* the term "linearized theory" for *all* conditions, static and nonstatic, to correspond to a set of uncontracted vortex tubes of constant pitch, with their vortex densities chosen such that $u'(0,r)$ is proportional to $\Gamma(r)$.

Results similar to these have already been reported in Ref. 6. We have repeated them here because our present solution is somewhat different and more accurate. In particular, both the Ψ interpolation scheme for computing the vortex tube shapes and the Gauss-Chebyshev integration schemes appear to be somewhat better than their counterparts in Ref. 6.

It is important to note that *for this case, where* $\Gamma(r) = \Gamma$ *and* $\lambda = 0$, *the streamtube pattern is virtually independent of* Γ over a wide range of Γ values. To see this analytically, we go back to the governing equations, Eqs. (8) and (10). For λ and hence $U = 0$, the Γ dependence cancels out of Eq. (8) since both $\Psi(0,R)$ and γ are proportional to γ_∞ which, in turn, contains the Γ dependence. From Eq. (10), we see that if we discard the swirl term $\Gamma^2/8\pi^2T^2$, γ will simply be proportional to $\sqrt{\Gamma}$ for $\lambda = 0$. With the swirl term omitted, it follows that the streamtube pattern will be *completely* independent of Γ although, of course, the velocities will scale proportional to $\sqrt{\Gamma}$. With the swirl term included, this result is no longer true in an exact sense. However, for a wide range of Γ's, $\Gamma^2/8\pi^2T^2 << \Omega\Gamma/2\pi$ in Eq. (10) — recall that Eq. (10) is in dimensional form — so our statement remains true in an approximate sense. To see this numerically, we re-ran Case 1 with Γ increased by a factor of ten and superimposed some typical streamtube points, indicated by small solid circles, on our original flow field in Fig. 6. They are seen to virtually coincide with the original streamtubes!

Next, we repeated Case 1 for the *nonstatic* cases $\lambda = 0.01$ and $\lambda = 0.10$. The speed of convergence improved slightly as λ increased. This is not

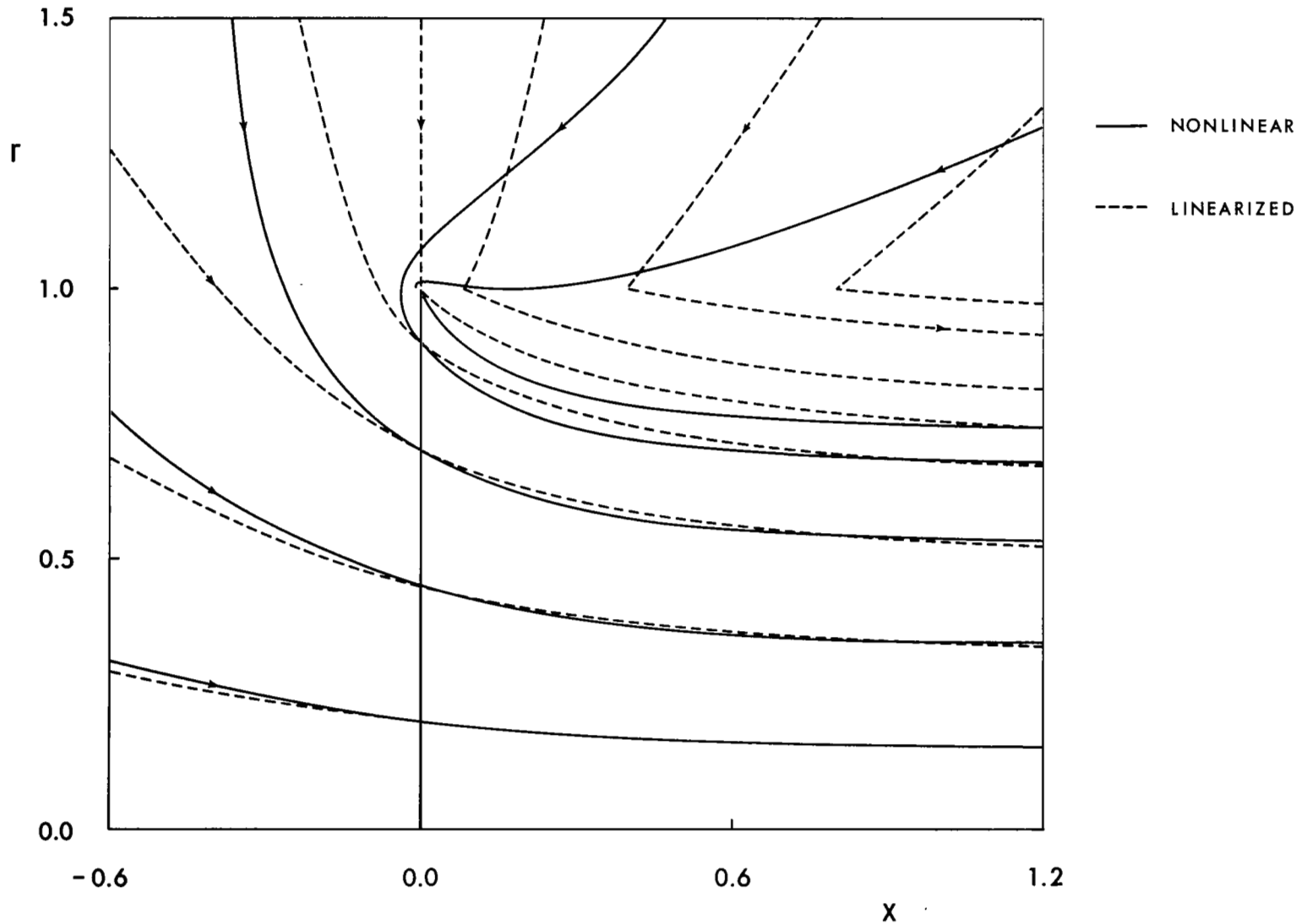

Figure 7. Comparison With Linearized Theory For Case 1, $\lambda = 0$

surprising because as $\lambda \to \infty$, for a fixed circulation distribution, the solution tends to the linearized solution* which, in fact, coincides with our zeroth iterate.

The results of the final converged iterations are as follows. For $\lambda = 0.01$, where four iterations were still needed,

$$a_{1j}^{(4)} = 0.13100,\ -2.24111,\ 17.34041,\ -79.79524,\ 222.82574,\ -355.64068,\ 317.69153,\ -148.11018,\ 28.55134$$

$$b_{1j}^{(4)} = -0.09605,\ 1.16135,\ -5.19913,\ 11.32189,\ -11.29768,\ 4.34854$$

and for $\lambda = 0.10$, where only three iterations were needed,

$$a_{1j}^{(3)} = 0.01933,\ -0.09949,\ 2.38665,\ -12.98060,\ 41.98347,\ -77.22364,\ 81.43538,\ -45.68608,\ 10.80515$$

$$b_{1j}^{(3)} = -0.00527,\ 0.28224,\ -1.22071,\ 2.85614,\ -2.97316,\ 1.16874$$

The final vortex densities are compared in Fig. 8 and the corresponding flow fields are shown in Figs. 9 and 10 for $\lambda = 0.01$ and 0.10 respectively.

Several striking results are apparent upon comparison of the three flow fields, Figs. 6, 9 and 10. First, we note a substantial change in the flow field as λ varies merely from 0.00 to 0.01 . For $\lambda = 0.01$ a "dividing streamtube", labeled DS in Figs. 9 and 10, appears, having the same Ψ value as the slipstream vortex tube. All fluid particles to the left of this streamtube eventually pass through the actuator disk and into the slipstream, whereas all other particles do not. Although we had anticipated that this streamtube would attach itself to the slipstream vortex tube at $x \approx 0.5$, our streamtube calculations seem to indicate that it turns downstream, perhaps joining the

---

* Physically, the pitch of the trailing vortices tends to infinity as $\lambda \to \infty$, so that $\gamma(x) \to 0$. Analytically, we see from Eq. (29) that $\gamma_\infty \sim F_\infty / 2\lambda^2 = O(1/\lambda^2)$ as $\lambda \to \infty$ for Γ fixed.

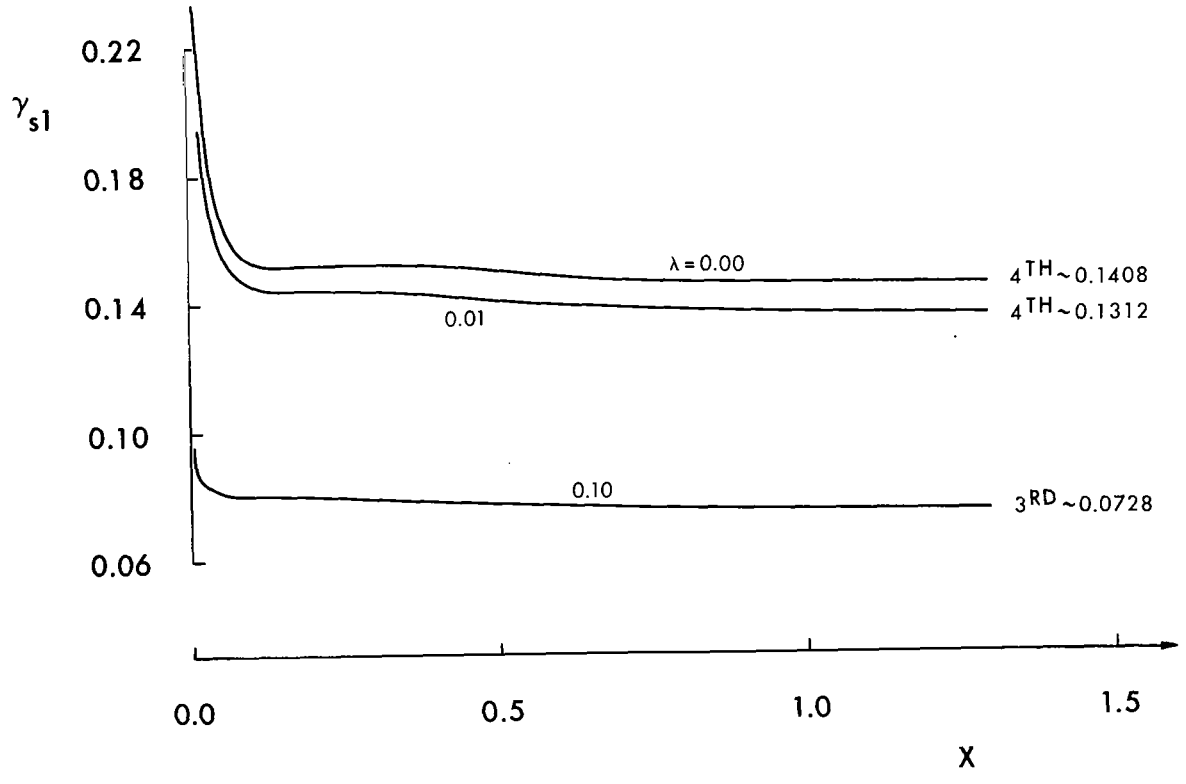

Figure 8. Comparison Of Converged Vortex Densities For Case 1, λ = 0.00, 0.01, 0.10

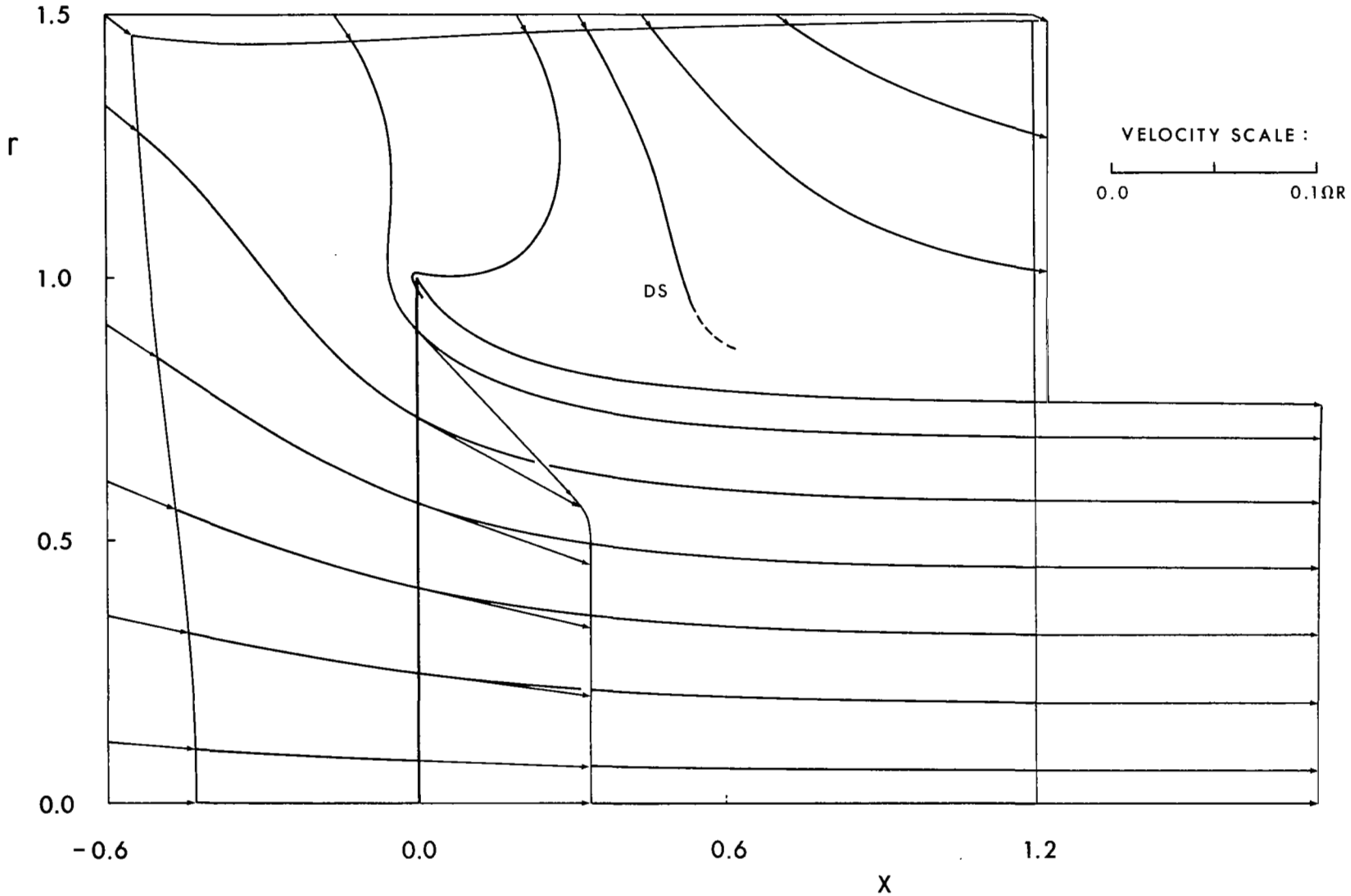

Figure 9. Streamtubes And Velocity Box For Case 1, $\lambda = 0.01$

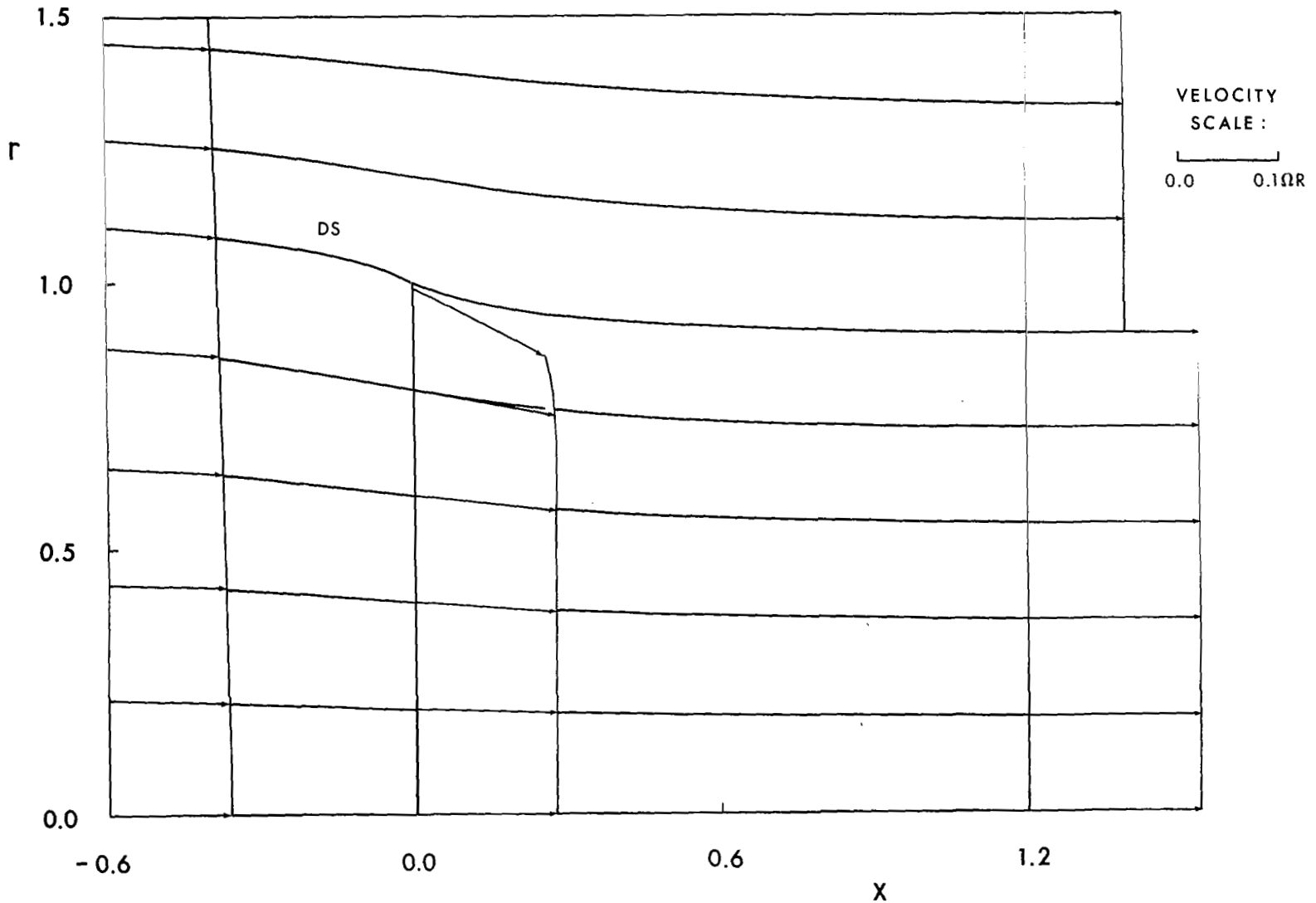

Figure 10. Streamtubes And Velocity Box For Case 1, $\lambda = 0.10$

vortex tube at $x = \infty$. In the absence of 100% numerical accuracy, we are not at present in a position to clarify this point. For $\lambda = 0.10$ the dividing streamtube almost blends smoothly with the slipstream vortex tube.

In static and low speed propeller testing, accurate performance measurements have been very difficult to obtain. That is, significant differences are found from test to test under seemingly identical conditions. Quite likely, though, there are small differences in the effective advance ratio due to recirculation within a closed test facility or due to ambient winds about an outside test facility. As a result, it is possible that the extreme sensitivity of the dividing streamtube to λ near $\lambda = 0$ and the associated changes in the fundamental nature of the flow may help to explain these measurement difficulties.

Finally, let us examine the nature of the flow in the important lip region. Theoretically, the flow around the lip persists for all finite λ's, disappearing only in the limiting case of $\lambda = \infty$. Practically, though, the lip flow has nearly disappeared at $\lambda = 0.10$ and is *almost* regular. For any given λ, then, the square root singularity in the vortex density $\gamma_{s1}(x)$ will be present, but its coefficient a_{11} will decrease rapidly as λ increases. For example, the a_{11}'s of the final iterates are 0.14676, 0.13100 and 0.01854 for $\lambda = 0.00$, 0.01 and 0.10 respectively.

Since the product of the vortex density γ_s and the meridional velocity ζ tends to a constant as $x \to 0^+$, as stated by Eq. (9), the singularity in γ_s implies that ζ must tend to zero as $x \to 0^+$. It follows that *the pitch of the trailing helical vortices tends to zero as* $x \to 0^+$, *even for* $\lambda > 0$, *causing the trailing vortices to linger in the disk plane!* This general behavior is in agreement with flow visualization studies and has an important bearing on the axial inflow over the outer portion of the blades and hence, on the blade design problem.

Whereas the meridional drift velocity *on* the slipstream tends to zero as $x \to 0^+$, the flow *around* the lip is singular, of course, due to the square root singularity in the vortex density. Thus, the axial inflow $u'(0,r)$ is singular as $r \to 1.0^-$, see Fig. 11. Notice that the singular flow is quite localized, being limited to the outer 2% of the disk for $\lambda = 0.00$ and 0.01 . For $\lambda = 0.10$

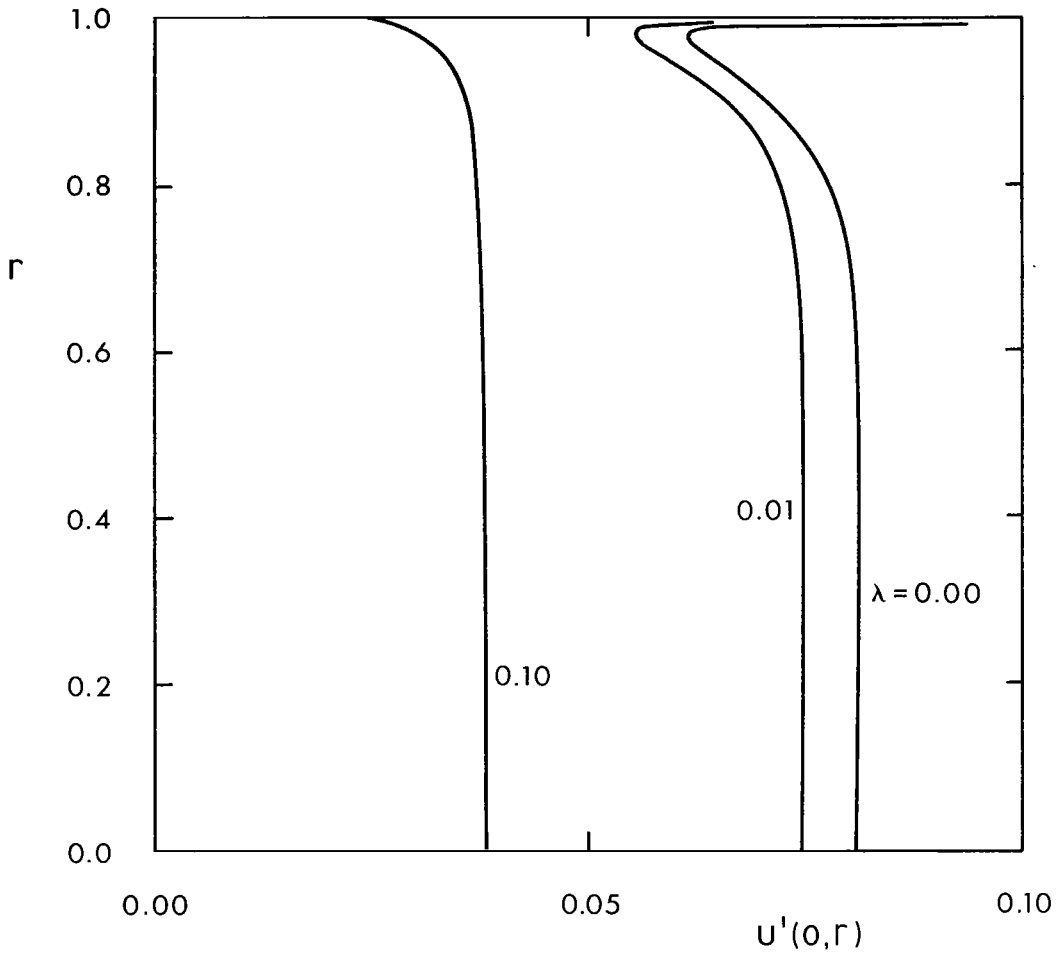

Figure 11. Axial Inflow Variation For Case 1, λ = 0.00, 0.01, 0.10

it is not even discernible in the plot. By way of comparison, the inflows predicted by linearized theory, e.g. Ref. 3, are proportional to $\Gamma(r)$ and hence *constant*.

Case 2 Results And Discussion. For our piecewise constant representation of the nonuniform circulation, as defined by Eq. (45) and Fig. 4, we took N, the β_ν's and the c_j's to be the same as in Case 1; N' = 6; the α_ν's = 0.02, 0.1, 0.3, 0.7, 1.2, 2.0; M = 7; the δ_ν's = 0.1, 0.2, 0.4, 0.7, 1.2, 2.0, 5.0; the $a_{kj}^{(0)}$'s, $b_{kj}^{(0)}$'s = 0 and NSWTH = 3, see Appendix A.3 .

We ran this case, through five iterations, for the same three advance ratios as in Case 1, namely, λ = 0.00, 0.01 and 0.10 . As in Case 1, excellent convergence was again obtained, although not quite as fast. For both λ = 0.00 and 0.01 the fourth and fifth iterates were virtually identical, with the second and third iterates in equally good agreement for λ = 0.10 .

Despite this success, though, we should hasten to point out that the convergence was very sensitive to whether Γ_1 is nonzero or not. We made several attempts to include a "cutout" or $\Gamma_1 = 0$, both with and without a singular vortex density, and were unable to achieve convergence. Physically, this is due to the fact that the innermost vortex tubes are situated in what is essentially a "dead air" or recirculatory flow region. With $\Gamma_1 \neq 0$, the axial velocity is positive everywhere within the slipstream and the problem is eliminated, albeit the details of the flow downstream near the axis are sacrificed and the remainder of the flow field is somewhat modified.

The a_{kj}'s of the final iterations are presented in Tables 1, 3 and 5 for λ = 0.00, 0.01 and 0.10, respectively. The corresponding b_{kj}'s are given in Tables 2, 4 and 6.

The final vortex densities are plotted for λ = 0.00, 0.01 and 0.10 in Figs. 12 - 14, respectively, and the corresponding flow fields in Figs. 15 - 17. The axial inflow u'(0,r) is plotted in Fig. 18. In Fig. 19 the streamlines are superimposed on the streamlines from the linearized theory of G. R. Hough and D. E. Ordway (Ref. 9).

Virtually all of our discussion of Case 1 applies equally well to Case 2. Two differences are apparent, however, upon comparison of the flow patterns; see, for example, Fig. 6 for Case 1 and Fig. 12 for Case 2, both with λ = 0.

TABLE 1. VORTEX DENSITY COEFFICIENTS a_{kj} FOR 5TH ITERATE
CASE 2, $\lambda = 0.00$

-5.31005	87.80304	-454.79934	1042.77150	-1020.40424	355.89366			
0.11810	-0.64985	0.92539	11.93209	-21.40161	10.21576			
0.19159	-2.33840	11.11373	-17.85558	12.54242	-2.82460			
0.13525	-1.65158	8.01838	-13.78819	10.46978	-2.46419			
0.09775	-1.13730	5.41043	-9.24617	6.86168	-1.34285			
0.02776	0.00785	-1.31852	5.89481	-8.77157	4.56386			
-0.02689	0.98278	-7.25287	20.66315	-24.98613	10.90950			
0.09994	-1.81353	22.48694	-170.80608	663.82119	-1341.73931	1440.63761	-776.87557	165.23356

TABLE 2. VORTEX TUBE SHAPE COEFFICIENTS b_{kj} FOR 5TH ITERATE
CASE 2, $\lambda = 0.00$

0.05161	-0.78685	5.08883	-15.86698	25.94582	-19.88912	5.52934
0.01628	-0.21965	1.54549	-4.88033	9.02835	-7.77928	2.37572
-0.00255	0.08792	-0.40993	1.37209	-1.11246	0.05569	0.11061
-0.00058	0.07347	-0.43210	1.86958	-2.50070	1.44753	-0.33873
0.00660	-0.02383	0.05808	0.74672	-1.28391	0.83792	-0.20402
0.05533	-0.72943	3.99908	-10.19713	14.63116	-10.91492	3.35289
0.15479	-2.24066	12.81023	-35.51269	52.86604	-40.11554	12.27317
0.37301	-5.60574	32.62229	-92.69815	139.47186	-106.28259	32.41277

TABLE 3. VORTEX DENSITY COEFFICIENTS a_{kj} FOR 5TH ITERATE
CASE 2, $\lambda = 0.01$

-1.75754	29.66592	-152.98610	362.90569	-359.78171	124.92775			
0.17389	-1.74082	8.09622	-7.26364	-0.31797	2.01773			
0.17604	-2.14618	10.92054	-19.11806	15.07162	-4.15971			
0.11808	-1.39736	7.21523	-13.07435	10.64075	-2.83983			
0.08265	-0.90922	4.62577	-8.29021	6.58843	-1.49248			
0.00864	0.23016	-1.92230	6.34250	-8.48524	4.23359			
-0.04620	1.13156	-7.09940	18.77593	-21.82192	9.35000			
0.09226	-1.35003	14.89530	-107.78903	422.62259	-867.39451	945.06865	-516.50995	111.40570

TABLE 4. VORTEX TUBE SHAPE COEFFICIENTS b_{kj} FOR 5TH ITERATE
CASE 2, $\lambda = 0.01$

0.02944	-0.45679	3.10690	-9.78935	16.56024	-13.08587	3.69735
-0.00255	0.04998	0.03410	-0.44590	2.22628	-2.73368	0.94900
-0.00987	0.17477	-0.80554	2.51764	-3.00991	1.56192	-0.33663
-0.00574	0.11874	-0.53751	2.06430	-2.86688	1.78849	-0.45220
0.00034	0.03180	-0.06996	0.90286	-1.47366	0.99641	-0.26013
0.03608	-0.49598	2.91415	-7.47287	10.75838	-8.02290	2.46645
0.11007	-1.62809	9.54367	-26.59581	39.72253	-30.19566	9.26035
0.27699	-4.19914	24.68756	-70.39518	106.22631	-81.15959	24.83005

TABLE 5. VORTEX DENSITY COEFFICIENTS a_{kj} FOR 3RD ITERATE
CASE 2, $\lambda = 0.10$

0.16124	-0.05617	1.05246	-0.13647	-0.79711	0.01866			
0.11348	-0.10530	1.07897	-1.07704	0.39278	-0.14728			
0.09587	-0.18512	1.33877	-2.12354	1.74859	-0.60581			
0.08332	-0.19473	1.26576	-2.22080	2.03598	-0.69016			
0.07653	-0.21103	1.24212	-2.31904	2.23864	-0.74307			
0.05075	0.06674	-0.44597	1.51121	-1.91270	0.99381			
0.03905	0.33492	-1.91290	4.86096	-5.41043	2.31422			
0.02080	-0.38777	4.56174	-18.63800	49.91276	-85.94686	90.69305	-52.12858	12.63717

TABLE 6. VORTEX TUBE SHAPE COEFFICIENTS b_{kj} FOR 3RD ITERATE
CASE 2, $\lambda = 0.10$

-0.00069	0.12995	-0.66299	2.00702	-3.01333	2.20558	-0.65428
0.00641	0.05556	-0.18654	0.68083	-1.07260	0.76284	-0.22515
0.00786	0.05871	-0.17408	0.65474	-1.04374	0.74826	-0.21939
0.01033	0.04072	-0.04899	0.30271	-0.52871	0.38212	-0.11404
0.01256	0.02444	0.05661	0.00597	-0.09730	0.08656	-0.03245
0.01829	-0.02246	0.33786	-0.81393	1.14311	-0.81561	0.24079
0.02575	-0.11497	0.88749	-2.45422	3.71240	-2.83834	0.88275
0.04066	-0.31024	2.04324	-5.83484	8.89683	-6.84617	2.12269

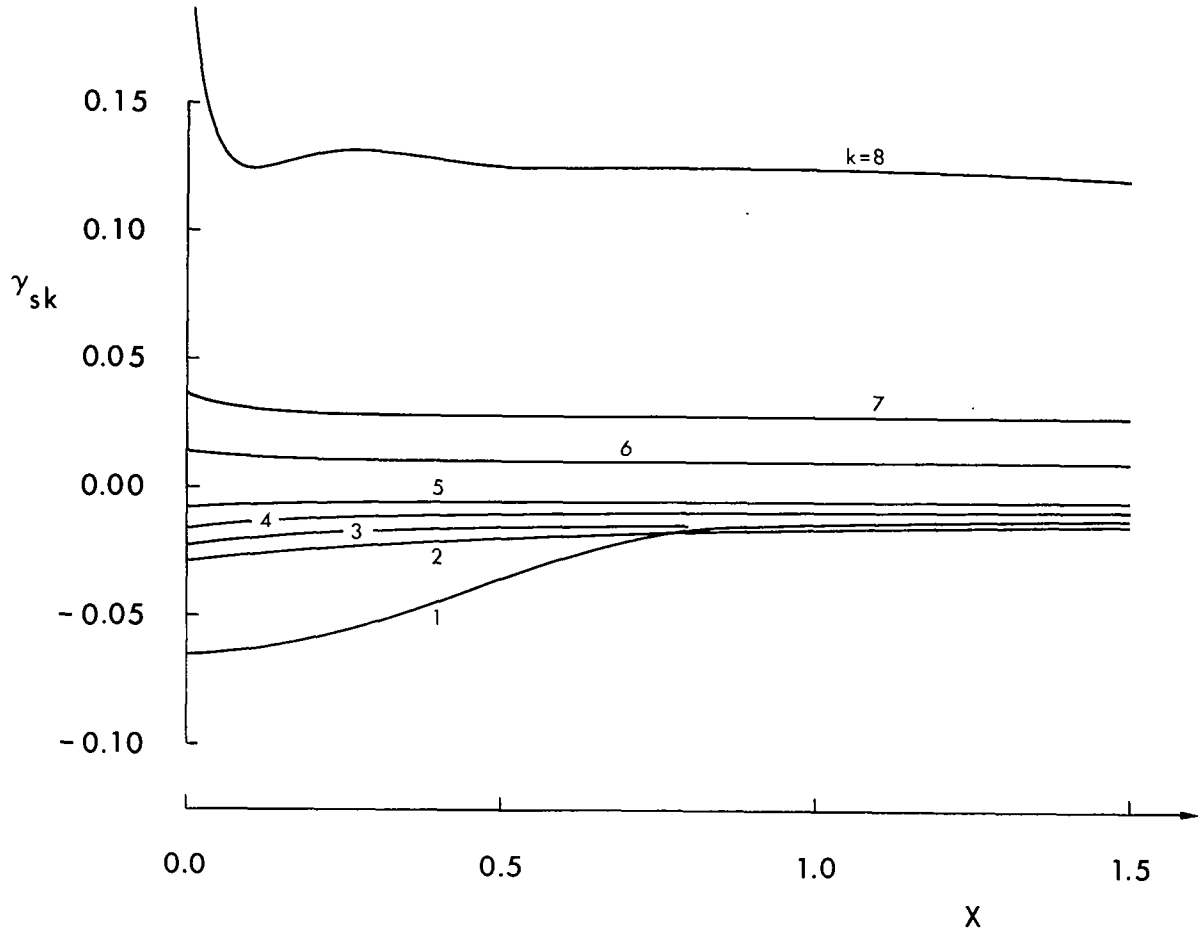

Figure 12. Vortex Densities Of 5th Iterate For Case 2, $\lambda = 0.00$

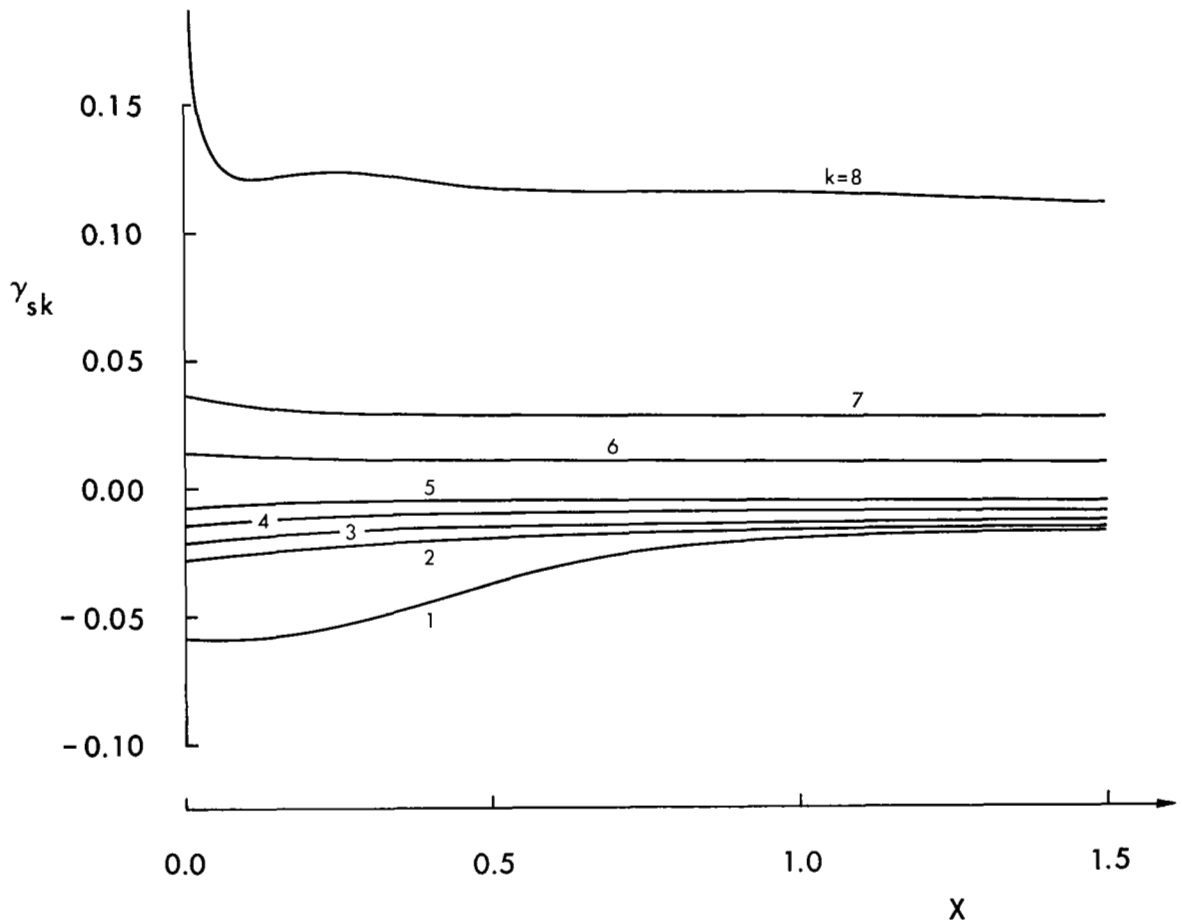

Figure 13. Vortex Densities Of 5th Iterate For Case 2, $\lambda = 0.01$

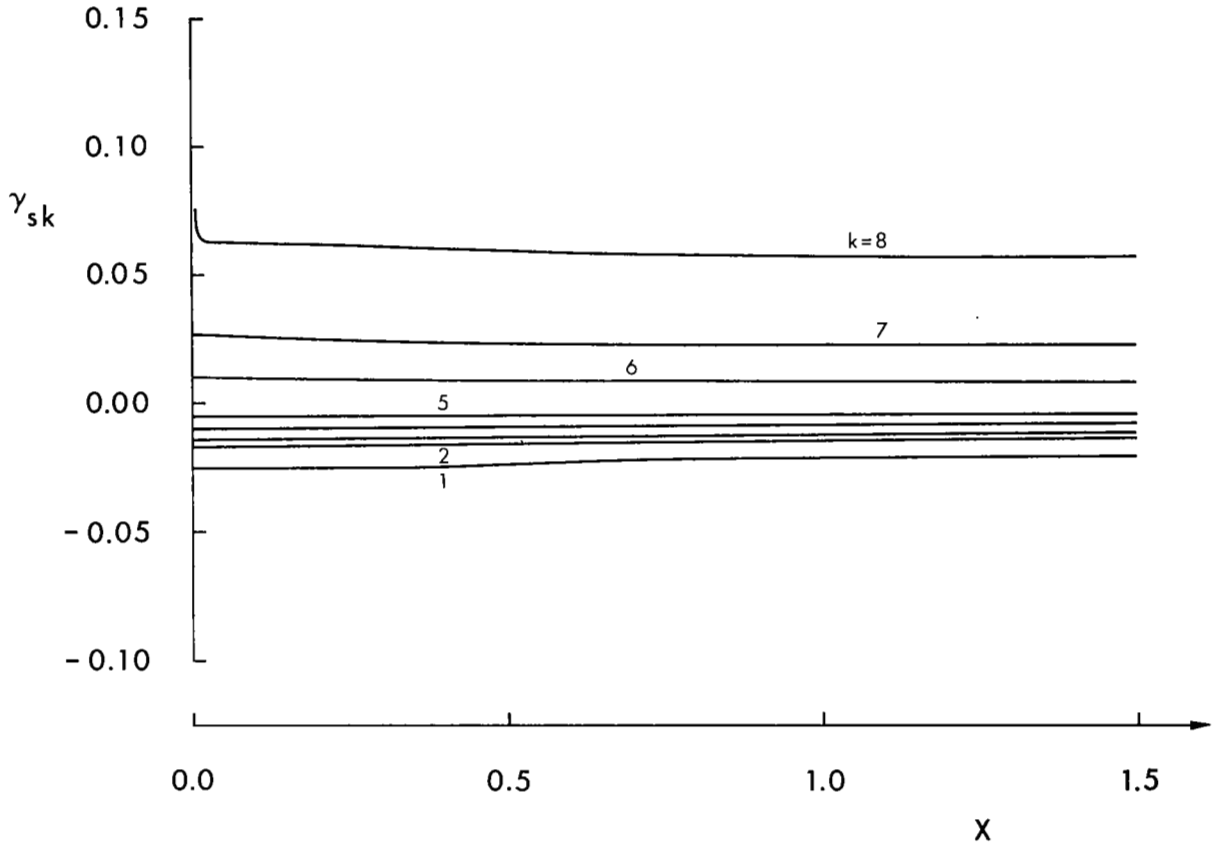

Figure 14. Vortex Densities Of 3rd Iterate For Case 2, $\lambda = 0.10$

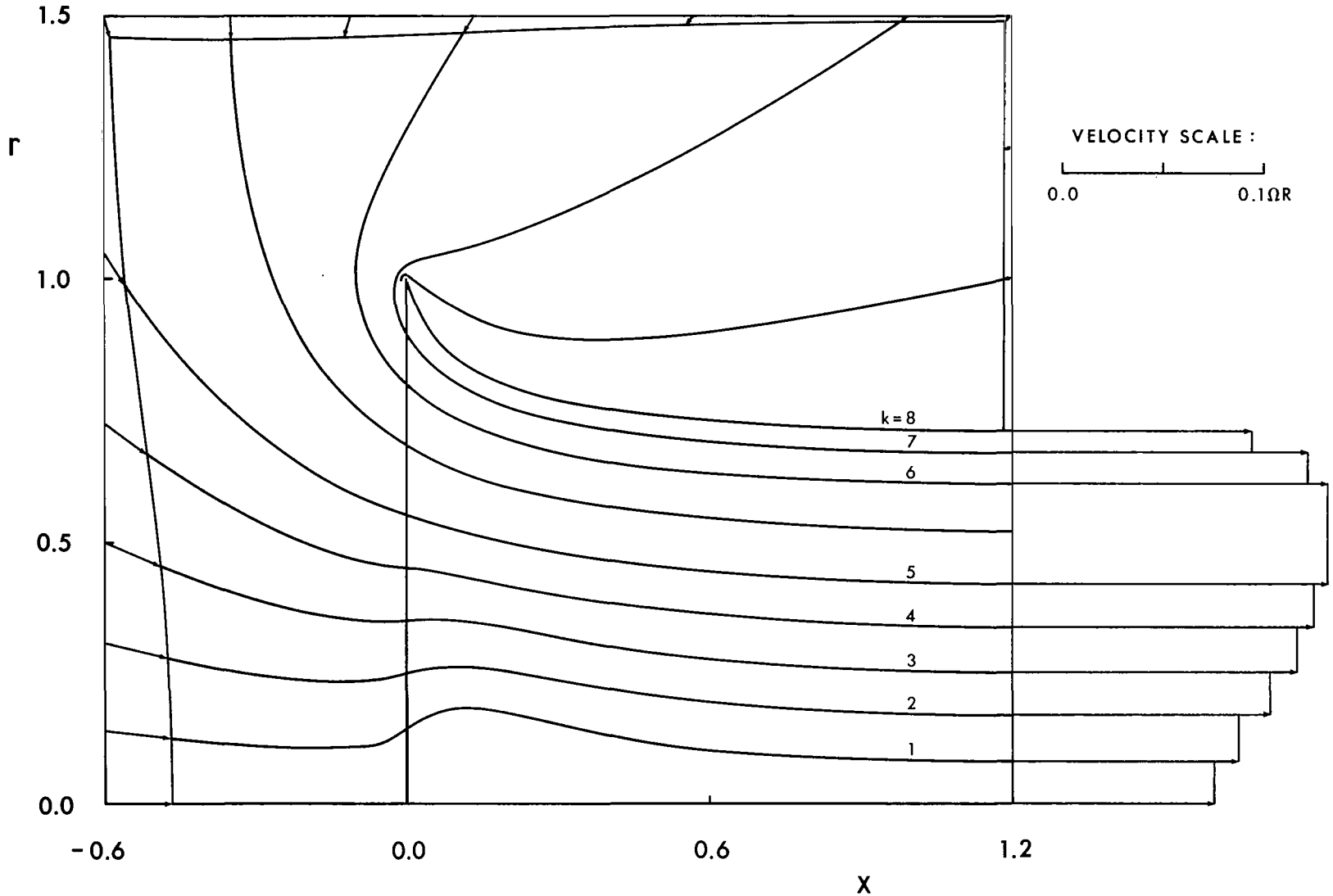

Figure 15. Streamtubes And Velocity Box For Case 2, $\lambda = 0.00$

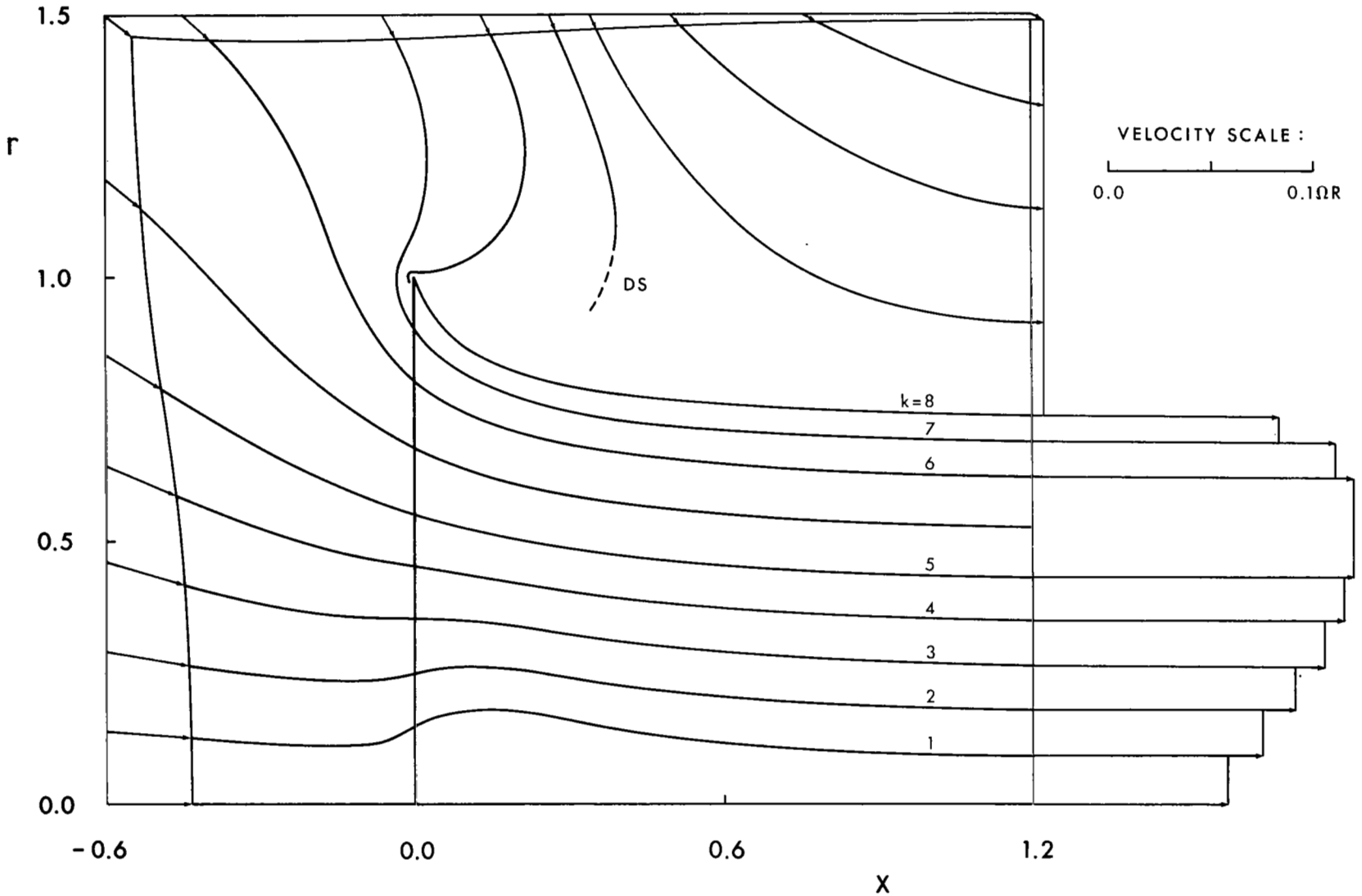

Figure 16. Streamtubes And Velocity Box For Case 2, $\lambda = 0.01$

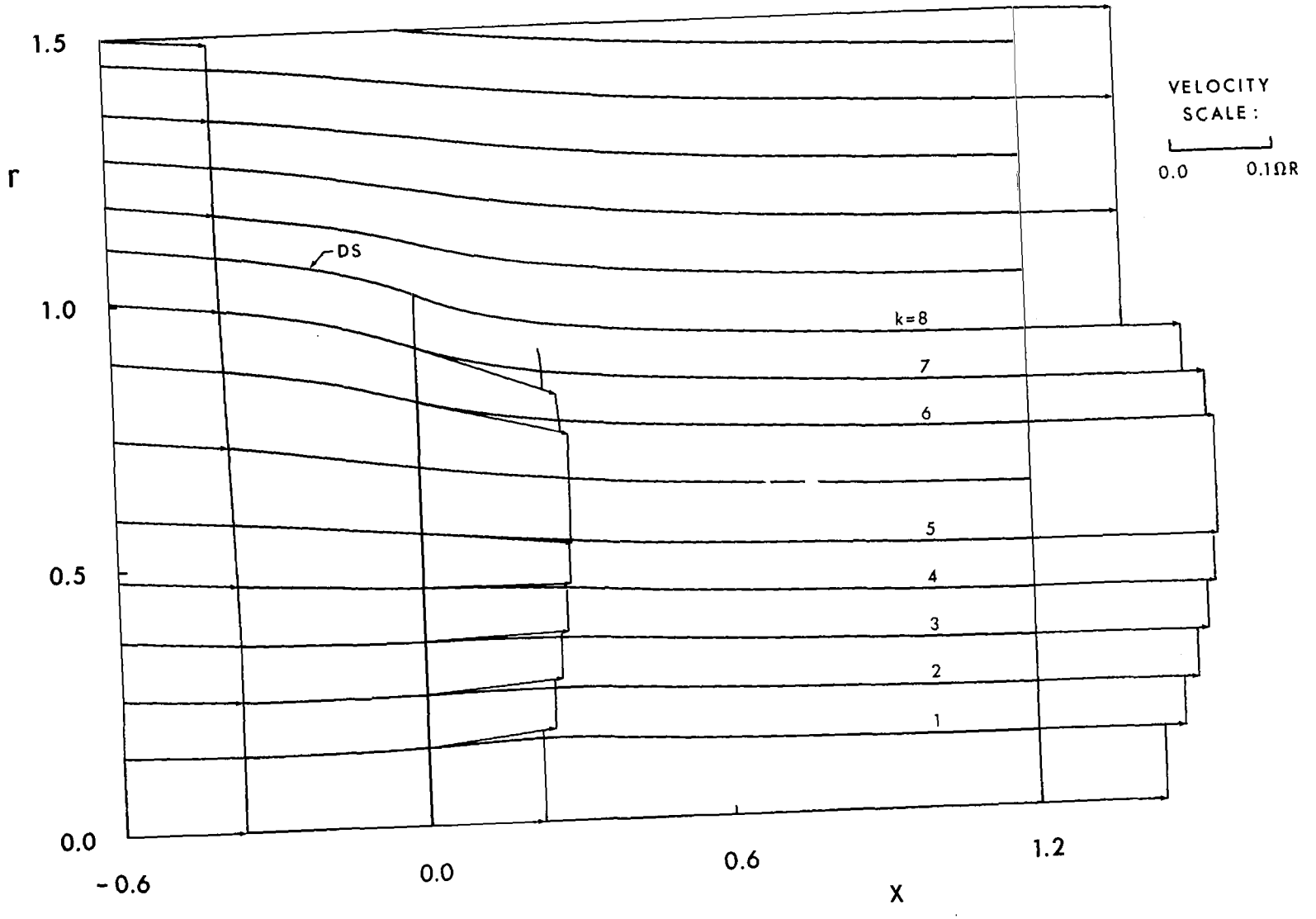

Figure 17. Streamtubes And Velocity Box For Case 2, $\lambda = 0.10$

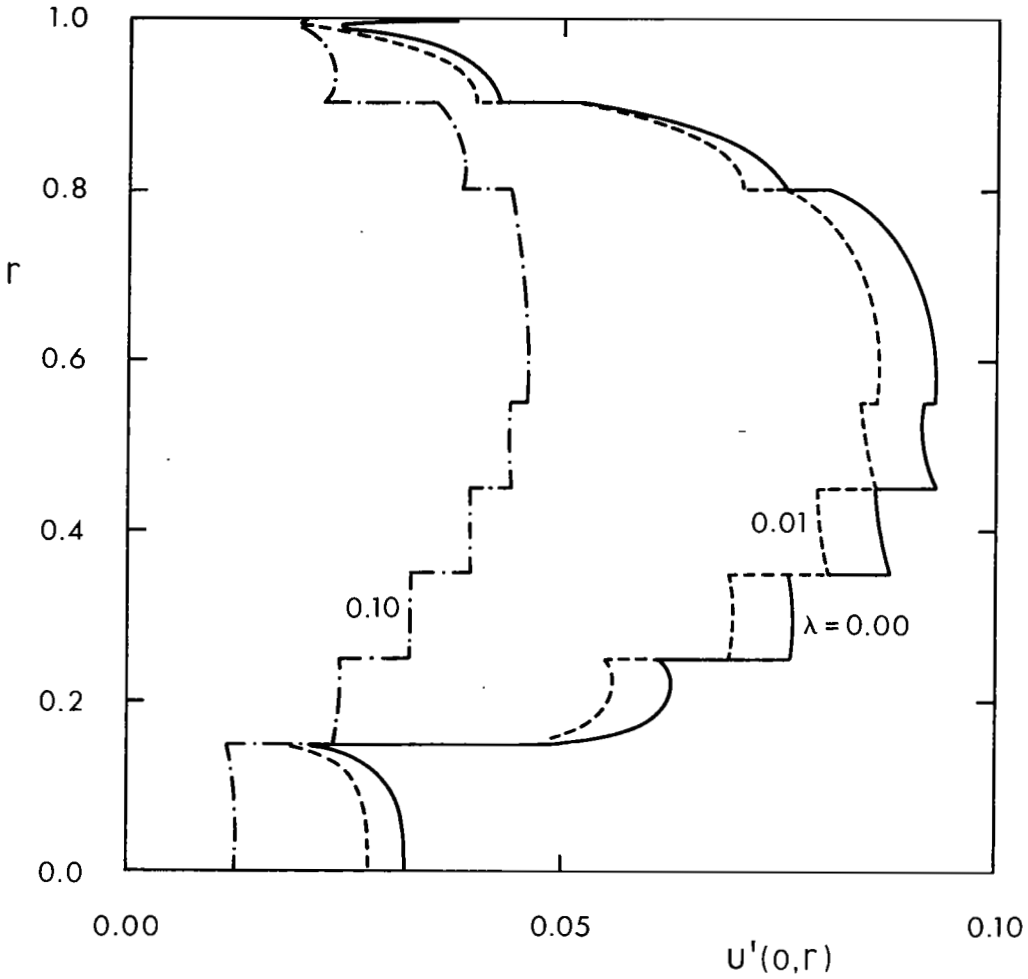

Figure 18. Axial Inflow Variation For Case 2, λ = 0.00, 0.01, 0.10

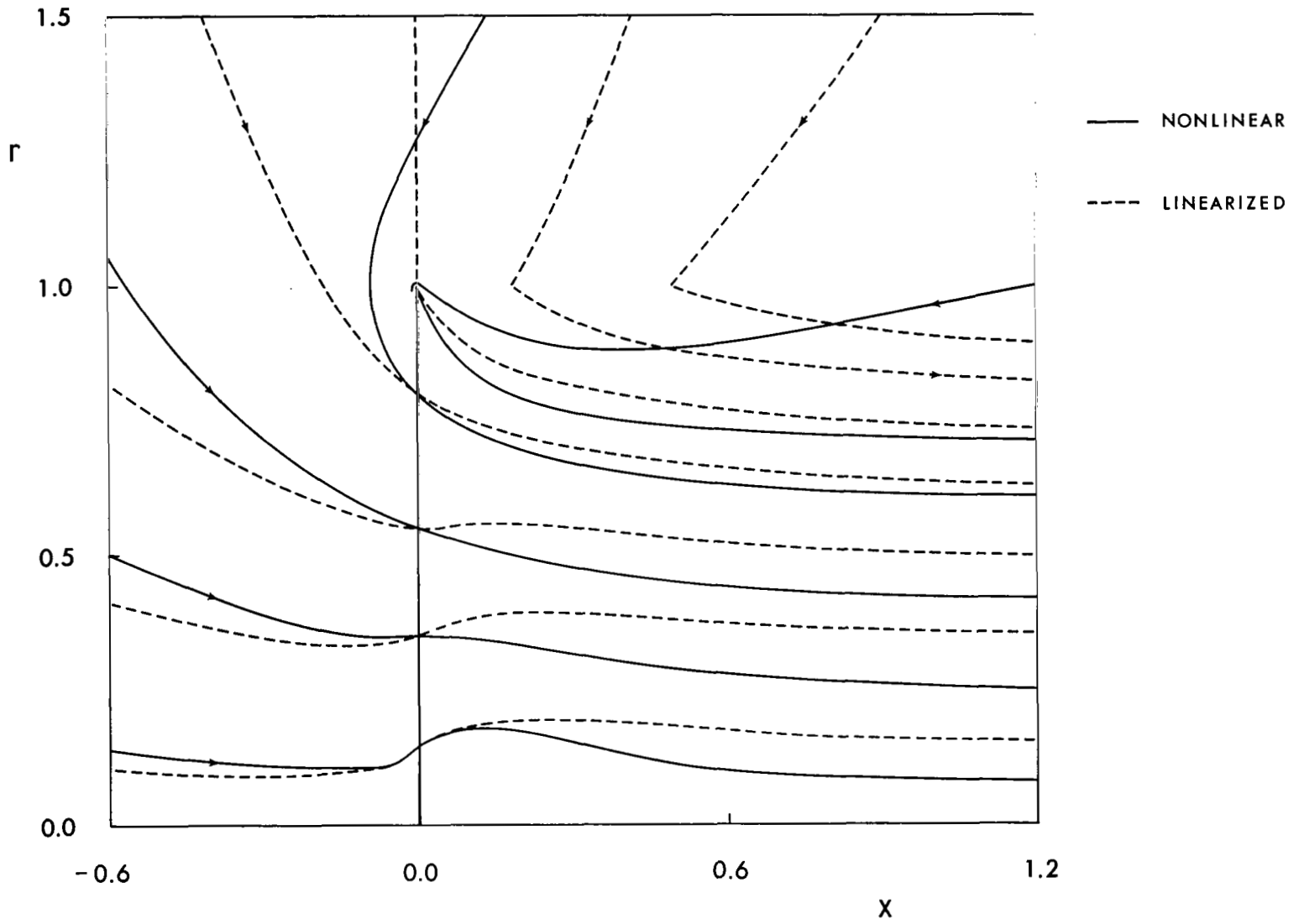

Figure 19. Comparison With Linearized Theory For Case 2, $\lambda = 0$

First, the *innermost vortex tubes expand initially* for Case 2, whereas the radial flow is purely negative *throughout* the slipstream for Case 1. This is due, of course, to the negative vorticity of the innermost vortex tubes and is predicted even by *linearized* theory — as seen in Fig. 19 for the static case. The expansion of the innermost vortex tubes is seen clearly, too, in the smoke pictures of J. B. Rorke and C. D. Wells (Ref. 10). In fact, a very close correlation is observed overall between our computed static flow field, Fig. 15, and the corresponding smoke picture in Fig. 3 of Ref. 10 for about the same C_T. Their actual circulation distribution, of course, and our prescribed piecewise constant distribution are undoubtedly somewhat dissimilar, but probably not enough to invalidate a comparison of the two flow fields. The worse dissimilarity most likely is the fact that $\Gamma_1 \neq 0$. As discussed above, therefore, we can not expect the flow field to be predicted properly along the axis in the slipstream. This is just what we find, Fig. 3 of Ref. 10 showing a trace of reverse flow.

Second, the *nonuniform circulation results in a sharper contraction of the slipstream*. At $x = 0.1$ and ∞, say, the contraction is 15.4% and 29.3% for Case 2, compared with 11.3% and 25.6% for Case 1. The reason appears to be as follows. Consider the outermost vortex tube $k = K = 8$ all by itself. Its self-induced contraction would be virtually the *same* as for Case 1. Now include the effect of the interior vortex tubes. The tubes for $k = 6$ and 7 carry positive vorticity and hence induce an additional contraction, whereas the other tubes, $k = 1, \ldots, 5$, carry negative vorticity and induce an expansion. The inward flow induced on the outermost tube by the tubes for $k = 6$ and 7 is much stronger, on the other hand, because of their close proximity to this tube, so that the *net* effect is an additional contraction. Numerically, these results are also in good accord with the experimental data of Rorke and Wells, at $x = 0.1$, see Fig. 6 of Ref. 10. They can not be compared exactly at $x = \infty$, but with extrapolation they would seem to agree except that the experimental values for the respective contractions are much nearer together. In line with our previous comments, this discrepancy may arise because $\Gamma_1 \neq 0$. That is, the presence of recirculatory flow along the slipstream axis would reduce the downstream contraction and so decrease the

differences in contraction between the cases for uniform and nonuniform circulation distributions.

We should make two further points. One concerns the axial velocity distribution within the slipstream. We see from Figs. 15, 16 and 18 that for small λ's this distribution is appreciably more uniform downstream than it is at the disk. The other point concerns the dependency of the streamtube pattern on the magnitude of the circulation distribution, or the thrust coefficient. We recall for the case of uniform circulation that the effects of swirl can be neglected and that the streamtube pattern is practically independent of Γ over a wide range of values. For Case 2, however, we found this is *not* true — at least as far as the inner flow is concerned, say within $r < 0.3$, where the swirl is strongest. To illustrate, we scaled our piecewise constant circulation distribution by a factor of five and the initial expansion of the innermost vortex tubes completely disappeared. Again, this could be related to the fact that $\Gamma_1 \neq 0$.

CONCLUSIONS

The axisymmetric flow field induced by an actuator disk with a prescribed nonuniform circulation distribution is considered. Coupled, nonlinear integral equations governing the wake geometry and the vortex density are developed from the force-free condition. These are discussed from both physical and mathematical points of view and an iterative solution is put forward.

Numerical results are presented, first for the case of a uniform circulation and then for a nonuniform circulation approximated by an eight-step, piecewise constant distribution. For each case, three different advance ratios, including the static condition are solved. All of these solutions are rapidly convergent for finite circulation at the hub axis.

The most important conclusions to be drawn from this study may be summarized as follows:

1. The flow around the lip of the outermost vortex tube trailing from the blade tips is singular for the static case and all finite advance ratios. As a result, the downstream drift velocity of the vorticity goes to zero at the lip and the strong trailing vortices emitted from the blade tips tend to

linger in the disk plane. This important feature of the flow, generally speaking, agrees with many flow visualization studies. It is only in the limit of infinite advance ratio that the singularity disappears.

2. Drastic changes in the flow pattern occur as the advance ratio is slightly increased from zero. Most pronounced of all is the sudden appearance of a "dividing streamtube", outside of the slipstream, which moves forward rapidly toward the lip with increasing advance ratio. It appears that this extreme sensitivity of the flow near the static condition may be the mechanism to help explain the difficulties of consistent propeller performance measurements at the static condition.

3. The streamtube pattern for the uniform circulation case is virtually independent of the magnitude of the disk loading. This is *not* true for the case of nonuniform circulation as far as the inner flow, out to about 30% of the blade radius, is concerned. This may be a consequence of the effect of swirl coupled with the finite value of circulation that we assumed for the first step in the distribution at the axis.

4. Comparison of the results for the uniform and the nonuniform cases, together with the observed nature of the interaction of adjacent vortex tubes, indicates that the faster the blade circulation drops off to zero at the blade tips, the less severe the slipstream contraction will be. These results are in good accord with experiment.

5. The computed flow patterns look quite reasonable and contain overall the main features observed in flow visualization studies. One feature that is lost, though, is the details of the dead air or recirculatory flow region in the slipstream along the axis. This is due to the finite value of circulation required at the axis, within the present numerical framework, for convergence.

APPENDIX A.1
ALTERNATE DERIVATION OF EQ. (3)

Whereas Wu's derivation of Eq. (3) proceeds directly from Lamb's form of the equations of motion, we would like to offer a somewhat different approach.

First, let us make a change of "meridional" variables, from (x,r) to (s,Ψ). We define the s variable so that the surfaces of constant s are orthogonal to the surfaces of constant Ψ. Thus,

$$\nabla\Psi\cdot\nabla s = \Psi_r s_r + \Psi_x s_x = urs_r - vrs_x = 0 \tag{A.1.1}$$

To relate ∂/∂s and ∂/∂Ψ to ∂/∂x and ∂/∂r, we write

$$\partial\Psi/\partial\Psi = 1 = \Psi_r r_\Psi + \Psi_x x_\Psi = urr_\Psi - vrx_\Psi \tag{A.1.2}$$

$$\partial s/\partial\Psi = 0 = s_r r_\Psi + s_x x_\Psi \tag{A.1.3}$$

Combining Eqs. (A.1.1) and (A.1.3), we have

$$vr_\Psi + ux_\Psi = 0, \tag{A.1.4}$$

and Eqs. (A.1.2) and (A.1.4) finally yield

$$\partial/\partial\Psi = (u\partial/\partial r - v\partial/\partial x) / r\zeta^2 \tag{A.1.5}$$

$$\partial/\partial s = (v\partial/\partial r + u\partial/\partial x) / \zeta \tag{A.1.6}$$

where ζ is the meridional velocity component, $\zeta = (u^2 + v^2)^{1/2}$.

Now, the circulation around an elemental meridional area dxdr is equal to the θ component of vorticity, $(v_x - u_r)$, times the area dxdr. By means of the Jacobian, and Eqs. (A.1.5), (A.1.6), (1) and (2), we have

$$(v_x - u_r)\ dxdr = -\ (\Psi_{rr} - \Psi_r/r + \Psi_{xx})\ dsd\Psi / r^2\zeta \tag{A.1.7}$$

since

$$v_x - u_r = -\ \Psi_{xx}/r - (\Psi_r/r)_r \tag{A.1.8}$$

and

$$dxdr = |\partial(x,r)/\partial(s,\Psi)|\ dsd\Psi$$

so

$$dxdr = |x_s r_\Psi - x_\Psi r_s| \; dsd\Psi$$

$$= dsd\Psi / r\zeta \qquad \text{(A.1.9)}$$

Alternatively, the differential of the circulation can be computed as γ_s ds, see Fig. 3b and the accompanying discussion. Eq. (9) gives γ_s for the special case $\Gamma = \Gamma(r)$, if we replace Γ and T by $-d\Gamma$ and r, respectively. Thus,

$$\gamma_s \; ds = (-d\Gamma/2\pi r) \; (\Omega r + w) \; ds / \zeta \qquad \text{(A.1.10)}$$

Equating Eqs. (A.1.7) and (A.1.10), and recalling that $\Gamma = -2\pi wr$, produces the desired result of Eq. (3).

APPENDIX A.2
NUMERICAL CALCULATION OF THE LEGENDRE FUNCTIONS

Our calculation of $Q_{\pm\frac{1}{2}}$ is based upon the relationship between $Q_{\pm\frac{1}{2}}$ and the complete elliptic integrals of the first and second kind, K and E, respectively, together with the remarkable approximations by C. Hastings (Ref. 11) for these functions*, each accurate to within $\pm\ 2 \times 10^{-8}$.

Using nested multiplication for maximum efficiency, we have, for all values of $z > 0$,

$$Q_{-\frac{1}{2}}(1+z) = \sqrt{2/(z+2)}\ \{A - B\,\ell n\, z'\} \tag{A.2.1}$$

$$Q_{+\frac{1}{2}}(1+z) = (1+z)\ Q_{-\frac{1}{2}}(1+z) - \sqrt{2(z+2)}\ \{C - D\,\ell n\, z'\} \tag{A.2.2}$$

where $z' = z/(z+2)$, and A, B, C and D are computed as follows,

$$\begin{aligned} a_1 &= 0.03742563713 + 0.01451196212z' \\ a_2 &= 0.03590092383 + a_1 z' \\ a_3 &= 0.09666344259 + a_2 z' \\ A &= 1.38629436112 + a_3 z' \end{aligned} \tag{A.2.3}$$

$$\begin{aligned} b_1 &= 0.03328355346 + 0.00441787012z' \\ b_2 &= 0.06880248576 + b_1 z' \\ b_3 &= 0.12498593597 + b_2 z' \\ B &= 0.5 + b_3 z' \end{aligned} \tag{A.2.4}$$

$$\begin{aligned} c_1 &= 0.04757383546 + 0.01736506451z' \\ c_2 &= 0.06260601220 + c_1 z' \\ c_3 &= 0.44325141463 + c_2 z' \\ C &= 1 + c_3 z' \end{aligned} \tag{A.2.5}$$

$$\begin{aligned} d_1 &= 0.04069697526 + 0.00526449639z' \\ d_2 &= 0.09200180037 + d_1 z' \end{aligned}$$

---

* We might mention that there is an error in the reproduction of Hastings' formulas for K and E in Ref. 12. Specifically, all of the m_1's in Eqs. 17.3.33 - 36 should be replaced by $(1 - m^2)$.

$$d_3 = 0.24998368310 + d_2 z'$$
$$\mathcal{D} = d_3 z' \qquad \text{(A.2.6)}$$

We would like to emphasize the numerical importance of treating the argument of $Q_{\pm\frac{1}{2}}$ as $(1+z)$ rather than simply $\tilde{\omega}$, say. The point is that $Q_{\pm\frac{1}{2}}(1+z) \sim -\frac{1}{2}\,\ell n\, z$ as $z \to 0$ so that if we compute first z and then $(1+z) = \tilde{\omega}$, to compute $\ell n\, z$ we must recompute $z = (\tilde{\omega}-1)$. For very small z's the process of adding unity and then subtracting it again leads to very sizeable round-off errors in z, and hence in $\ell n\, z$, and hence in $Q_{\pm\frac{1}{2}}(1+z)$.

APPENDIX A.3
NUMERICAL INTEGRATION SCHEMES

<u>$\Psi(x,r)$ Calculations In MAIN CODE</u>. Consider the calculation of any of the integrals, say I_ν, which occur in the stream function,

$$\Psi(x,r) = \lambda r^2/2 + \sum_{\nu=1}^{K} \int_0^\infty G(\xi,t_\nu;x,r)\ \gamma_\nu(\xi)\ d\xi$$

$$\equiv \lambda r^2/2 + \sum_{\nu=1}^{K} I_\nu(x,r) \qquad \text{(A.3.1)}$$

For $x = 0$, the singularities of the integrand are as follows: $G = O(\ell n\ \xi)$ as $\xi \to 0$ if $r = R_\nu$, but is regular otherwise, and $\gamma_\nu = O(\xi^{-\frac{1}{2}})$ as $\xi \to 0$ if $\nu = K$, but is regular otherwise. Depending on these circumstances, the integrand is either $O(\xi^{-\frac{1}{2}}\ \ell n\ \xi)$, $O(\ell n\ \xi)$ or $O(1)$ as $\xi \to 0$; in any event it is $O(\xi^{-3})$ as $\xi \to \infty$. Therefore, we first introduce a new variable μ defined by $\xi = (\mu^{1.1} + \mu^3)$. The effect of the μ^3 term is to compress the "tail" portion so that we can, with very good accuracy, replace the upper limit $\xi = \infty$ by $\mu = 5$, say. The effect of the $\mu^{1.1}$ term is to alter the behavior at the origin to $O(\mu^{-0.45}\ \ell n\ \mu)$, $O(\mu^{-0.45})$ or $O(1)$. Since the *worst* possible singularity is now *weaker* than $\mu^{-\frac{1}{2}}$, the μ integration is well suited to *Gauss-Chebyshev* integration, which is capable of coping with singularities up to square root strength at one or both end points. For this integration scheme, we then set $\mu = 2.5(\tau+1)$ to change the interval to $-1 \le \tau \le 1$ and obtain,

$$I_\nu(0,r) \approx \frac{2.5\pi}{L} \sum_{\ell=1}^{L} F\left[\cos \frac{(2\ell-1)\pi}{2L}\right] \qquad \text{(A.3.2)}$$

where

$$F(\tau) = \sqrt{1-\tau^2}\ G[\xi,t_\nu(\xi);0,r]\ (1.1\mu^{0.1} + 3\mu^2)\ \gamma_\nu(\xi) \qquad \text{(A.3.3)}$$

For $x > 0$, we split the ξ integral into two parts, 0 to x and x to ∞, and proceed along lines similar to the $x = 0$ case for each of these parts. This gives

$$I_\nu(x,r) \approx \frac{\pi}{2L} \sum_{\ell=1}^{L} \left\{ xF_1 \left[\cos \frac{(2\ell-1)\pi}{2L} \right] + 5F_2 \left[\cos \frac{(2\ell-1)\pi}{2L} \right] \right\} \qquad (A.3.4)$$

taking F_1 and F_2 as

$$F_1(\tau) = \sqrt{1-\tau^2}\ G[\xi, t_\nu(\xi); x, r]\ \gamma_\nu(\xi)$$

$$F_2(\tau) = \sqrt{1-\tau^2}\ G[\xi, t_\nu(\xi); x, r]\ (1+3\mu^2)\ \gamma_\nu(\xi) \qquad (A.3.5)$$

with the change of variables here of $\xi = x(\tau+1)/2$ in F_1 and $\xi = (x+\mu+\mu^3)$, $\mu = 2.5(\tau+1)$ in F_2.

From experience, we use L = 80 in Eq. (A.3.2) if $r \neq R_\nu$, and L = 160 if $r = R_\nu$. In Eq. (A.3.4), we use L = 40 if r is *not* on $t_\nu(\xi)$, and L = 80 if it *is*.

As an example of the accuracy thus achieved, we note that with K = 1, $T_1(x) = 1.0$, $\gamma_1(x) = 1.0$ and $\lambda = 0$, for example, we compute $\Psi(0,1) = 0.25002$ compared with an exact value of 0.25 .

Actually, it is uneconomical to require such a high level of accuracy for *all* steps of the iteration, since the first couple of iterates are not particularly close to the exact solution anyway. As a consequence, we have introduced an integer input variable NSWTH into our MAIN CODE. For all iterations < NSWTH we let L = 20 for *all* integrals in the MAIN CODE, including those in the vortex density calculation, to be discussed below, and for all iterations ≥ NSWTH we switch to the more accurate scheme. By way of comparison, for the above case with K = 1, $T_1(x) = 1.0$, $\gamma_1(x) = 1.0$ and $\lambda = 0$, we compute $\Psi(0,1) = 0.25179$ for L = 20, compared with the more accurate value of 0.25002 for L = 160.

<u>All Vortex Density Integrals In MAIN CODE</u>. For any one of the integrals involved in the calculation of the vortex density, say,

$$I(x,r) = \int_0^\infty F(\xi; x, r)\ d\xi$$

we utilize the following Gauss-Chebyshev scheme,

$$I(x,r) \approx \frac{\pi}{2L} \sum_{\ell=1}^{L} F\left[\cos \frac{(2\ell-1)\pi}{2L}\right] \tag{A.3.6}$$

where

$$F(\tau) = \sqrt{1-\tau^2} \left\{ x\,F[0.5x(\tau+1);x,r] + \right.$$

$$\left. x\,F[0.5x(\tau+3);x,r] + 15\mu^2 F[(2x+\mu^3);x,r] \right\} \tag{A.3.7}$$

with $\mu = 2.5(\tau+1)$ and $L = 20$ for all iterations < NSWTH, and $L = 40$ for all iterations $\geq$ NSWTH.

<u>$\Psi(x,r)$ Calculations In STREAMTUBE CODE</u>. For $x \leq 0$ we compute our Ψ's according to Eqs. (A.3.1) - (A.3.3), and for $x > 0$ according to Eqs. (A.3.1), (A.3.4) and (A.3.5). If $|x| < 0.05$, we use $L = 160$ and if $|x| \geq 0.05$, we use $L = 80$.

<u>Velocity Integrals In VELOCITY BOX CODE</u>. To determine the velocity components, we have to evaluate the expressions,

$$u(x,r) = \lambda + \frac{1}{r} \sum_{k=1}^{K} \int_0^\infty G_r(\xi,t_k;x,r)\ \gamma_k\ d\xi \quad , \quad r > 0$$

$$= \lambda + \frac{1}{2} \sum_{k=1}^{K} \int_0^\infty t_k^2[(\xi-x)^2 + t_k^2]^{-3/2}\ \gamma_k\ d\xi \quad , \quad r = 0$$

$$v(x,r) = -\frac{1}{r} \sum_{k=1}^{K} \int_0^\infty G_x(\xi,t_k;x,r)\ \gamma_k\ d\xi \quad , \quad r > 0$$

$$= 0 \quad , \quad r = 0 \tag{A.3.8}$$

In these equations, G_r and G_x may be derived from Eq. (5) with the help of Eq. (13). We find that G_r is identical to G_T, see Eqs. (14) and (15), if we replace t by t_k and T by r and interpret all lengths in their nondimensional form. For G_x we obtain

$$G_x(\xi,t_k;x,r) = \frac{x-\xi}{4\pi(\tilde{\omega}_2^2 - 1)\sqrt{rt_k}} [\tilde{\omega}_2 Q_{\frac{1}{2}}(\tilde{\omega}_2) - Q_{-\frac{1}{2}}(\tilde{\omega}_2)] \tag{A.3.9}$$

in which the argument $\tilde{\omega}_2$ is identical to $\tilde{\omega}$ of Eq. (6) with ρ replaced by t_k.

Instead of a Gauss-Chebyshev scheme for the integrals appearing in Eqs. (A.3.8), better accuracy was achieved by splitting each integral into thirteen parts, from 0 to 0.001, 0.001 to 0.01, 0.01 to 0.1, 0.1 to 0.5, 0.5 to 1.1, 1.1 to 1.19, 1.19 to 1.2, 1.2 to 1.21, 1.21 to 1.3, 1.3 to 2.0, 2.0 to 10, 10 to 50, and 50 to 200, and then using a ten point Gauss-Legendre formula on each part.

As a measure of the accuracy of this approach, consider again the simple special case $K = 1$, $T_1(x) = 1.0$, $\gamma_1(x) = 1.0$ and $\lambda = 0$. We compute $u(0,0)$, $u(0,0.5)$, $u(0,0.9)$, $u(0,0.99)$ and $u(0,0.999)$ all equal to 0.499992, compared with the exact value 0.5, and $u(0,1.0) = 0.24995$, compared with the exact value 0.25 .

APPENDIX A.4
COMPUTER CODES AND LISTINGS

Description Of MAIN CODE. The MAIN CODE, written in Fortran IV for the IBM 1130 Computer at Sage Action, Inc., contains the iterative solution for the shapes $T_k^{(n)}(x)$ and vortex densities $\gamma_k^{(n)}(x)$, as described in the foregoing THEORETICAL DEVELOPMENT.

The physical input consists of λ, K, the R_k's, and the Γ_k's for k = 1 through K, where $R_K = 1.0$. Important non-physical input includes N and the associated β_ν's and c_j's, N' and the associated α_ν's, M and the associated δ_ν's, and the initial vortex density and shape coefficients $a_{kj}^{(0)}$ and $b_{kj}^{(0)}$. Generally, all of the $a_{kj}^{(0)}$'s, $b_{kj}^{(0)}$'s = 0, as implied by Eqs. (34) and (35), but this need not be the case.

Maximum allowable values of K, N, N' and M are $K \leq 18$, $N \leq 11$, $N' \leq 8$ and $M \leq 10$.

As a measure of the computing cost, the machine time per iteration for Cases 1 and 2 was approximately 10 minutes and 3 hours respectively, with the more accurate final iterations taking about twice as long. This is not as bad as it sounds, however, since the IBM 1130 is a relatively slow machine, with a correspondingly modest hourly cost.

Finally, as a word of caution, we note that convergence of the iteration depends somewhat on the choice of the β_ν's, c_j's and δ_ν's. The values used in Cases 1 and 2 were arrived at essentially by trial, but they seem to work well for a variety of other cases as well.

Description Of STREAMTUBE CODE. The purpose of the STREAMTUBE CODE is to compute the streamtube pattern corresponding to the vortex wake configuration obtained from the MAIN CODE.

Basically, we first compute $\Psi(x,r)$ at 25 or less specified (x,r) "field points", and then search for those Ψ values by interpolation throughout a network of computed values. The "interpolation net" is broken into two regions. Region 1 consists of the intersection of the prescribed x and r values; 20 or less negative x's, say from 0.0 back to -0.6, and 25 or less r's, say from 0.0 up to 1.5 — not necessarily evenly spaced. Region 2 is defined by 27 or less positive x's, say from 0.0 to 1.2, and 20 or less evenly spaced

r's, from $T_K(x)$ up to some prescribed constant such as 1.5 . Since Region 1 is rectangular, we use three point interpolation in both x *and* r there, whereas we interpolate only in r in Region 2 because of the nonlinear slipstream boundary. The interpolated locations are then printed out.

To trace these streamtubes in the *slipstream* region, $x \geq 0$ and $0 \leq r < T_K(x)$, we usually interpolate by hand from the interpolation net printed out in the last iteration of the MAIN CODE. For the special case of $K = 1$, though, we can simply rerun the STREAMTUBE CODE with the constant r boundary of Region 2 replaced by $r = 0$ and avoid excessive duplication of the upstream calculation by use of only one (x,r) point in Region 1.

Besides the field points and net points, the input consists of λ, N, N', M and K and the R_k's, Γ_k's, a_{kj}'s, b_{kj}'s and c_j's.

Description Of VELOCITY BOX CODE. The VELOCITY BOX CODE simply computes the meridional velocity $(u^2 + v^2)^{\frac{1}{2}}$ and its inclination $\tan^{-1}(v/u)$ at arbitrarily many (x,r) field points on the propeller disk, $x = 0$ and $0 \leq r < 1$, and on the "box" defined by $x = -0.6$ and $0 \leq r \leq 1.5$, $-0.6 \leq x \leq 1.2$ and $r = 1.5$, and $x = 1.2$ and $0 \leq r \leq 1.5$.

The input consists of the (x,r) field points, plus λ, N, N', M and K and the R_k's, Γ_k's, a_{kj}'s, b_{kj}'s and c_j's.

All three codes use extended precision.

Computer Code Listings. Listings of the three computer codes are reproduced on the following pages.

```
C     M A I N   C O D E
C
C     M P R O G
C
C THE MAIN CODE CONSISTS OF THREE DISTINCT PROGRAMS LINKED TOGETHER -
C
C     MPROG IS THE PROGRAM INITIALLY EXECUTED.  IT DIRECTS ALL INPUT AND
C     INITIALIZATION.  UPON COMPLETION, A LINK IS TAKEN TO SPROG.
C
C     SPROG CONTROLS THE SHAPE CALCULATION PORTION OF THE MAIN CODE.  UPON
C     COMPLETION OF EACH ITERATION, A LINK IS TAKEN TO GPROG.
C
C     GPROG DIRECTS THE VORTEX DENSITY CALCULATION OF THE MAIN CODE.  UPON
C     COMPLETION OF EACH ITERATION, THE MOST RECENT SET OF COEFFICIENTS
C     ARE STORED.  IF MORE ITERATIONS ARE TO BE PERFORMED, A LINK IS TAKEN
C     BACK TO SPROG.
C
C ALL INPUT FOR THE MAIN CODE IS DONE IN SUBROUTINE INITL.
C INPUT VARIABLES ARE AS FOLLOWS -
C
C VARIABLE NAME      DESCRIPTION
C -------------      ------------------------------------------------------------
C            IM      TWO DIGIT MONTH OF THE YEAR
C            ID      TWO DIGIT DAY OF THE MONTH
C            IY      TWO DIGIT YEAR
C          KASE      TEN CHARACTER CASE IDENTIFICATION
C         LAMBA      ADVANCE RATIO
C            NN      NUMBER OF COLLOCATION POINTS IN SOLUTION OF VORTEX DENSITY
C                         FOR OUTER VORTEX TUBE - ALSO NUMBER OF TERMS IN
C                         VORTEX DENSITY FUNCTION FOR OUTER VORTEX TUBE
C            NP      NUMBER OF COLLOCATION POINTS IN SOLUTION OF VORTEX DENSITY
C                         FOR INNER VORTEX TUBES - ALSO NUMBER OF TERMS IN
C                         VORTEX DENSITY FUNCTION FOR INNER VORTEX TUBES
C            MM      NUMBER OF AXIAL STATIONS USED IN STREAM FUNCTION
C                         INTERPOLATION - ALSO NUMBER OF TERMS IN STREAM
C                         FUNCTION
C            KK      NUMBER OF STEPS IN THE PIECEWISE CONSTANT CIRCULATION
C                         DISTRIBUTION
C         ITBEG      BEGINNING ITERATION NUMBER
C         ITEND      ENDING ITERATION NUMBER
C         NPLOT      NUMBER OF ITERATIONS TO BE PLOTTED ON ONE GRID
C         NSWTH      NUMBER OF ITERATION AT WHICH INTEGRATION ACCURACY IS TO
C                         BE INCREASED
C            RR      RADIAL LOCATIONS OF STEPS IN CIRCULATION DISTRIBUTION
C         CAPGM      PIECEWISE CONSTANT CIRCULATION DISTRIBUTION
C          BETA      AXIAL STATIONS USED IN SOLVING FOR THE VORTEX DENSITY
C                         ON THE OUTER VORTEX TUBE
C         ALPHA      AXIAL STATIONS USED IN SOLVING FOR THE VORTEX DENSITIES
C                         ON ALL BUT THE OUTER VORTEX TUBE
C         DELTA      AXIAL STATIONS USED IN STREAM FUNCTION INTERPOLATION
C             C      EXPONENTS IN MATCHING FUNCTION FOR VORTEX DENSITY ON
C                         OUTER VORTEX TUBE
C             B      COEFFICIENTS IN REPRESENTATION OF VORTEX TUBE SHAPE
C             A      COEFFICIENTS IN REPRESENTATION OF VORTEX DENSITY
C
C NOTES ON INPUT -
C 1  IF KK=1, NP AND BETA NEED NOT BE SPECIFIED
C 2  IF ITBEG=0, A AND B COEFFICIENTS ARE SET TO ZERO
C
```

```
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
      DEFINE FILE 1(18,32,U,IREC1), 2(36,33,U,IREC2), 3(1,320,U,IREC3),
     *            4(1,320,U,IREC4), 5(10,320,U,IREC5)
      CALL INITL
      CALL GAMIN
      ITERT=ITBEG-1
      CALL BPLOT
      DO 10 K=1,KK
      CALL SPLOT(K)
   10 CONTINUE
      DO 20 K=1,KK
      CALL GPLOT(K)
   20 CONTINUE
      CALL EPLOT(1,5.0,0.0)
      CALL LINK(SPROG)
      END


C     S P R O G
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
      DEFINE FILE 1(18,32,U,IREC1), 2(36,33,U,IREC2), 3(1,320,U,IREC3),
     *            4(1,320,U,IREC4), 5(10,320,U,IREC5)
      CALL SCALE(1.0,1.0,5.0,0.0)
      ITERT=ITERT+1
      CALL BPLOT
      CALL PSIN1
      CALL PSIN2
      DO 10 K=1,KK
      CALL REGIN(K)
      CALL ESOLV(TRHS,TLHS,B,MM)
      WRITE(1'K)(B(J),J=1,MM)
      CALL SPLOT(K)
   10 CONTINUE
      CALL EPLOT(1,5.0,0.0)
      CALL LINK(GPROG)
      END


C     G P R O G
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
      DEFINE FILE 1(18,32,U,IREC1), 2(36,33,U,IREC2), 3(1,320,U,IREC3),
     *            4(1,320,U,IREC4), 5(10,320,U,IREC5),
     *            6(180,32,U,IREC6), 7(180,33,U,IREC7)
      CALL SCALE(1.0,1.0,5.0,0.0)
      CALL GAMIN
      KLES1=KK-1
      IF(KLES1)40,30,10
   10 DO 20 K=1,KLES1
      CALL GAMM1(K)
```

```
      CALL ESOLV(TRHS,TLHS,A,NP)
      K2=18+K
      WRITE(2'K2)(A(J),J=1,NP)
      CALL GPLOT(K)
   20 CONTINUE
   30 CALL GAMM2(KK)
      CALL ESOLV(TRHS,TLHS,A,NN)
      K2=18+KK
      WRITE(2'K2)(A(J),J=1,NN)
      CALL GPLOT(KK)
   40 DO 50 K=1,KK
      K2=18+K
      READ(2'K2)(A(J),J=1,NN)
      WRITE(2'K)(A(J),J=1,NN)
   50 CONTINUE
      CALL PUTCF
      IF(ITERT-ITEND)60,70,70
   60 CALL EPLOT(1,5.0,0.0)
      CALL LINK(SPROG)
   70 CALL EPLOT(1,18.0,0.0)
      CALL EXIT
      END


      SUBROUTINE PSIN1
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C EQUATION 23
      WRITE(3,3000)
      CALL TAUFT(1,L1)
      CALL TAUFT(2,L2)
      DO 40 K=1,KK
      PSINF(K)=0.0
      DO 30 NU=1,KK
      IF(NU-K)20,10,20
   10 CALL ONINT(TAU1,L1,RR(K),NU,SUM)
      PSINF(K)=PSINF(K)+SUM
      GO TO 30
   20 CALL ONINT(TAU2,L2,RR(K),NU,SUM)
      PSINF(K)=PSINF(K)+SUM
   30 CONTINUE
      PSINF(K)=LAMBA*RR(K)**2/2.0+PSINF(K)
      WRITE(3,3001)RR(K),PSINF(K)
   40 CONTINUE
      RETURN
 3000 FORMAT(//' PSI AT X = 0 AND R = R SUB K'/)
 3001 FORMAT(10X,'PSI(0,',F4.2,') =',E20.11)
      END


      SUBROUTINE ONINT(TAU,LL,R,K,SUM)
      DIMENSION TAU(1)
C INTEGRATION FOR EQUATION 23
      SUM=0.0
      DO 10 L=1,LL
      S1=2.5*(1.0+TAU(L))
      S2=2.5*(1.0-TAU(L))
      X1=S1**1.1+S1*S1*S1
```

```
      X2=S2**1.1+S2*S2*S2
      T1=TTFCT(K,X1)
      T2=TTFCT(K,X2)
      G1=GGFCT(X1,T1,0.0,R)
      G2=GGFCT(X2,T2,0.0,R)
      TERM1=1.1*S1**0.1+3.0*S1*S1
      TERM2=1.1*S2**0.1+3.0*S2*S2
      GAM1=GMFCT(2,K,X1)
      GAM2=GMFCT(2,K,X2)
      ROOT=SQRT(1.0-TAU(L)**2)
      F1=ROOT*G1*TERM1*GAM1
      F2=ROOT*G2*TERM2*GAM2
      SUM=SUM+F1+F2
   10 CONTINUE
      SUM=SUM*3.926990816/LL
      RETURN
      END


      SUBROUTINE PSIN2
      REAL LAMBA
      DIMENSION TT(37),PSI(37)
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C EQUATION A3.1
      WRITE(3,3000)
      CALL TAUFT(2,L2)
      CALL TAUFT(3,L3)
      TT(1)=0.0
      PSI(1)=0.0
      DO 50 J=1,MM
      DO 20 I=2,NPSI
      PSI(I)=0.0
      IF(I-2*(I/2))20,10,20
   10 TT(I+1)=TTFCT((I+1)/2,DELTA(J))
      TT(I)=(TT(I+1)+TT(I-1))/2.0
   20 CONTINUE
      DO 30 K=1,KK
      CALL TWINT(TAU2,L2,DELTA(J),K,TT(2*K+1),SUM)
      PSI(2*K+1)=PSI(2*K+1)+SUM
      CALL THINT(TAU3,L3,DELTA(J),K,TT,PSI,NPSI)
   30 CONTINUE
      DO 40 I=2,NPSI
      PSI(I)=LAMBA*TT(I)**2/2.0+PSI(I)
   40 CONTINUE
      WRITE(3,3001)DELTA(J),(TT(I),PSI(I),I=1,NPSI)
      WRITE(5'J)(PSI(I),TT(I),I=1,NPSI)
   50 CONTINUE
      RETURN
 3000 FORMAT(///' PSI NET FOR INTERPOLATION'//14X,1HX,9X,1HR,12X,8HPSI(X
     *,R))
 3001 FORMAT(5X,2F10.5,E20.10/(15X,F10.5,E20.10))
      END


      SUBROUTINE TWINT(TAU,LL,X,K,R,SUM)
      DIMENSION TAU(1)
C INTEGRATION FOR EQUATION A3.1 WHEN R IS ON VORTEX TUBE K
      SUM=0.0
```

```
      DO 10 L=1,LL
      X1=X*(1.0+TAU(L))/2.0
      X2=X*(1.0-TAU(L))/2.0
      T1=TTFCT(K,X1)
      T2=TTFCT(K,X2)
      SUM=SUM+X/2.0*SQRT(1.0-TAU(L)**2)*(GGFCT(X1,T1,X,R)*GMFCT(2,K,X1)+
     *                                   GGFCT(X2,T2,X,R)*GMFCT(2,K,X2))
      S1=2.5*(1.0+TAU(L))
      S2=2.5*(1.0-TAU(L))
      X1=X+S1+S1*S1*S1
      X2=X+S2+S2*S2*S2
      T1=TTFCT(K,X1)
      T2=TTFCT(K,X2)
      SUM=SUM+2.5*SQRT(1.0-TAU(L)**2)*
     *         (GGFCT(X1,T1,X,R)*(1.0+3.0*S1*S1)*GMFCT(2,K,X1)+
     *          GGFCT(X2,T2,X,R)*(1.0+3.0*S2*S2)*GMFCT(2,K,X2))
   10 CONTINUE
      SUM=SUM*1.5707963268/LL
      RETURN
      END


      SUBROUTINE THINT(TAU,LL,X,K,TT,PSI,NPSI)
      DIMENSION TAU(1),TT(1),PSI(1)
C INTEGRATION FOR EQUATION A3.1 WHEN R IS NOT ON VORTEX TUBE K
      DO 10 I=2,NPSI
      PSI(I)=PSI(I)*LL/1.5707963268
   10 CONTINUE
      DO 40 L=1,LL
      S3=2.5*(1.0+TAU(L))
      S4=2.5*(1.0-TAU(L))
      X1=X*(1.0+TAU(L))/2.0
      X2=X*(1.0-TAU(L))/2.0
      X3=X+S3+S3*S3*S3
      X4=X+S4+S4*S4*S4
      T1=TTFCT(K,X1)
      T2=TTFCT(K,X2)
      T3=TTFCT(K,X3)
      T4=TTFCT(K,X4)
      GAM1=GMFCT(2,K,X1)
      GAM2=GMFCT(2,K,X2)
      GAM3=GMFCT(2,K,X3)
      GAM4=GMFCT(2,K,X4)
      ROOT=SQRT(1.0-TAU(L)**2)
      TERM1=ROOT*0.5*X*GAM1
      TERM2=ROOT*0.5*X*GAM2
      TERM3=ROOT*(2.5+7.5*S3*S3)*GAM3
      TERM4=ROOT*(2.5+7.5*S4*S4)*GAM4
      DO 30 I=2,NPSI
      IF(2*K+1-I)20,30,20
   20 R=TT(I)
      PSI(I)=PSI(I)+GGFCT(X1,T1,X,R)*TERM1+GGFCT(X2,T2,X,R)*TERM2+
     *              GGFCT(X3,T3,X,R)*TERM3+GGFCT(X4,T4,X,R)*TERM4
   30 CONTINUE
   40 CONTINUE
      DO 50 I=2,NPSI
      PSI(I)=PSI(I)*1.5707963268/LL
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE REGIN(K)
      REAL LAMBA
      DIMENSION TT(37),PSI(37)
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C SHAPE INTERPOLATION FOR EQUATION 36
      READ(3'1)((TRHS(J,NU),NU=1,MM),J=1,MM)
      DO 60 J=1,MM
      READ(5'J)(PSI(L),TT(L),L=1,NPSI)
      DO 10 NREGN=3,NPSI,2
      IF(PSI(NREGN)-PSINF(K))10,20,20
   10 CONTINUE
      GAMSM=0.0
      DO 15 KSUM=K,KK
      GAMSM=GAMSM+GMFCT(2,KSUM,BETA(J))
   15 CONTINUE
      TLHS(J)=SQRT(TTFCT(K,DELTA(J))**2+2.0*(PSINF(K)-PSI(2*K+1))/
     *                (LAMBA+GAMSM*(DELTA(J)+0.15)/(DELTA(J)+0.30)))-RR(K)
      GO TO 50
   20 T1=TT(NREGN)
      T2=TT(NREGN-1)
      T3=TT(NREGN-2)
      P1=PSI(NREGN)
      P2=PSI(NREGN-1)
      P3=PSI(NREGN-2)
      CCOEF=(P3-P1+(T1-T3)*(P1-P2)/(T1-T2))/ (T1*T2-T1*T3-T2*T3+T3*T3)
      BCOEF=(P1-P2)/(T1-T2)-CCOEF*(T1+T2)
      ACOEF=P1-BCOEF*T1-CCOEF*T1*T1
      TPLUS=SQRT(BCOEF*BCOEF-4.0*CCOEF*(ACOEF-PSINF(K)))
      TMINS=-(BCOEF+TPLUS)/(2.0*CCOEF)
      TPLUS=-(BCOEF-TPLUS)/(2.0*CCOEF)
      IF(T1-TPLUS)30,40,25
   25 IF(TPLUS-T3)30,40,40
   30 TLHS(J)=TMINS-RR(K)
      GO TO 50
   40 TLHS(J)=TPLUS-RR(K)
   50 CONTINUE
      NREGN=NREGN/2
   60 CONTINUE
      RETURN
      END


      FUNCTION GGFCT(XI,T,XX,TT)
C EQUATION 5
      TERM1=TT*T
      DEL=(XX-XI)**2
      Z=((TT-T)**2+DEL)/(2.0*TERM1)
      CALL QQPMH(Z,QPH,QMH)
      GGFCT=SQRT(TERM1)/6.2831853072*QPH
      RETURN
      END


      SUBROUTINE SPLOT(K)
      REAL LAMBA
      DIMENSION XW(11),TTW(11)
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
```

```
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C PRINT AND PLOT ROUTINE FOR VORTEX TUBE SHAPE
      XW(1)=0.01
      DO 8 J=1,10
      IF(J-MM)4,4,6
    4 XW(J+1)=DELTA(J)
      GO TO 8
    6 XW(J+1)=-2.0**125
      TTW(J)=-2.0**125
    8 CONTINUE
      KGO=ITERT-ITBEG+1
      KGO=KGO-(NPLOT+1)*(KGO/(NPLOT+1))
      KGO=K-1+((KGO+2)*(KGO+3))/2
      KPEN=1
      DO 40 I=1,101
      XC=(I-1.0)/50.0
      YC=TTFCT(K,XC)
      IF(YC)30,10,10
   10 IF(1.0-YC)30,20,20
   20 XP=10-XC*5.0
      YP=YC*5.0
      CALL EPLOT(KPEN,YP,XP)
      KPEN=2
      IF(I-KGO*(I/KGO))40,30,40
   30 KPEN=1
   40 CONTINUE
      IF(K-1)50,50,60
   50 WRITE(3,3001)(XW(I),I=1,10)
   60 MPLUS=MM+1
      DO 70 I=1,MPLUS
      TTW(I)=TTFCT(K,XW(I))
   70 CONTINUE
      TKINF=TTFCT(K,1000.0)
      WRITE(3,3002)K,(TTW(I),I=1,10),TKINF,(B(J),J=1,MM)
      WRITE(3,3003)
      RETURN
 3001 FORMAT(///12X,1HX,2X,10F9.3,5X,4HINF./)
 3002 FORMAT(8H ' T SUB,I2,5X,11F9.5/2H '/11H ' B COEFFS,9X,5E18.10/
     *(2H ',18X,5E18.10))
 3003 FORMAT(2H '/2H ')
      END


      SUBROUTINE GAMM1(K)
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C EQUATION 41
      CALL TAUFT(3,L3)
      DO 20 J=1,NP
      TT=TTFCT(K,ALPHA(J))
      FAC=SQRT(1.0+TPFCT(K,ALPHA(J))**2)
      FF=(CAPGM(K)-CAPGM(K+1)-(CAPGM(K)**2-CAPGM(K+1)**2)/
     *     (12.5663706144*TT**2))/(6.2831853072*GMINF(2,K)*FAC)
      TLHS(J)=0.0
      DO 10 NU=1,KK
      CALL FOINT(0,TAU3,L3,ALPHA(J),TT,NU,SUM)
      TLHS(J)=TLHS(J)+SUM
```

```
   10 CONTINUE
      TLHS(J)=FF/(LAMBA+TLHS(J)/TT)-1.0
   20 CONTINUE
      READ(4'1)((TRHS(J,NU),NU=1,NP),J=1,NP)
      RETURN
      END


      SUBROUTINE GAMM2(K)
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C EQUATION 43
      CALL TAUFT(3,L3)
      DO 60 J=1,NN
      TT=TTFCT(K,BETA(J))
      FAC=SQRT(1.0+TPFCT(K,BETA(J))**2)
      FF=(CAPGM(K)-CAPGM(K+1)-(CAPGM(K)**2-CAPGM(K+1)**2)/
     *     (12.5663706144*TT**2))/3.1415926536
      GAMMA=GMFCT(1,K,BETA(J))
      CALL FOINT(J,TAU3,L3,BETA(J),TT,K,SUM)
      TLHS(J)=(FF/(2.0*GAMMA*GAMMA)*(2.0*GAMMA-GMINF(2,K)*FAC)-LAMBA)
     *         *TT-GMINF(2,K)*SUM
      IF(KK-1)40,40,10
   10 DO 30 NU=1,KK
      IF(NU-K)20,30,20
   20 CALL FOINT(0,TAU3,L3,BETA(J),TT,NU,SUM)
      TLHS(J)=TLHS(J)-SUM
   30 CONTINUE
   40 TLHS(J)=TLHS(J)/GMINF(2,K)
      TERM=FF*TT*FAC/(2.0*GAMMA*GAMMA)
      DO 50 I=1,NN
      TRHS(J,I)=TERM*EXP(-3.0*BETA(J))*BETA(J)**C(I)+TRHS(J,I)
   50 CONTINUE
   60 CONTINUE
      RETURN
      END


      SUBROUTINE FOINT(I,TAU,LL,X,R,K,SUM)
      REAL LAMBA
      DIMENSION TAU(1)
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C INTEGRATION FOR EQUATIONS 41 AND 43
      FAC0(K,X0)=SQRT(1.0+TPFCT(K,X0)**2)
      FUNC0(TAU0,FT1,FT2,FT3,FT4,FT5,FT6)=SQRT(1.0-TAU0*TAU0)*
     *          (FT1+FT2+FT3+FT4+FT5+FT6)
      FUNC1(F0,X0,C0)=F0*EXP(-3.0*X0)*X0**C0
      IF(I)30,30,10
   10 DO 20 J=1,NN
      TRHS(I,J)=0.0
   20 CONTINUE
   30 SUM=0.0
      XOVR2=X/2.0
      DO 70 L=1,LL
      TAUL=TAU(L)
```

```
      S3=2.5*(1.0+TAUL)
      S6=2.5*(1.0-TAUL)
      X1=XOVR2*(1.0+TAUL)
      X2=XOVR2*(3.0+TAUL)
      X3=2.0*X+S3*S3*S3
      X4=XOVR2*(1.0-TAUL)
      X5=XOVR2*(3.0-TAUL)
      X6=2.0*X+S6*S6*S6
      T1=TTFCT(K,X1)
      T2=TTFCT(K,X2)
      T3=TTFCT(K,X3)
      T4=TTFCT(K,X4)
      T5=TTFCT(K,X5)
      T6=TTFCT(K,X6)
      F1=XOVR2*GTFCT(X1,T1,X,R)
      F2=XOVR2*GTFCT(X2,T2,X,R)
      F3=7.5*S3*S3*GTFCT(X3,T3,X,R)
      F4=XOVR2*GTFCT(X4,T4,X,R)
      F5=XOVR2*GTFCT(X5,T5,X,R)
      F6=7.5*S6*S6*GTFCT(X6,T6,X,R)
      IF(I)40,40,50
   40 F1=F1*GMFCT(1,K,X1)
      F2=F2*GMFCT(1,K,X2)
      F3=F3*GMFCT(1,K,X3)
      F4=F4*GMFCT(1,K,X4)
      F5=F5*GMFCT(1,K,X5)
      F6=F6*GMFCT(1,K,X6)
      SUM=SUM+FUNC0(TAUL,F1,F2,F3,F4,F5,F6)
      GO TO 70
   50 F1=F1*FAC0(K,X1)
      F2=F2*FAC0(K,X2)
      F3=F3*FAC0(K,X3)
      F4=F4*FAC0(K,X4)
      F5=F5*FAC0(K,X5)
      F6=F6*FAC0(K,X6)
      SUM=SUM+FUNC0(TAUL,F1,F2,F3,F4,F5,F6)
      DO 60 J=1,NN
      T1=FUNC1(F1,X1,C(J))
      T2=FUNC1(F2,X2,C(J))
      T3=FUNC1(F3,X3,C(J))
      T4=FUNC1(F4,X4,C(J))
      T5=FUNC1(F5,X5,C(J))
      T6=FUNC1(F6,X6,C(J))
      TRHS(I,J)=TRHS(I,J)+FUNC0(TAUL,T1,T2,T3,T4,T5,T6)
   60 CONTINUE
   70 CONTINUE
      SUM=SUM*1.5707963268/LL
      IF(I)100,100,80
   80 DO 90 J=1,NN
      TRHS(I,J)=TRHS(I,J)*1.5707963268/LL
   90 CONTINUE
  100 RETURN
      END


      FUNCTION GTFCT(XI,T,XX,TT)
C EQUATION 14
      TERM1=TT*T
      DEL=(XX-XI)**2
      Z=((TT-T)**2+DEL)/(2.0*TERM1)
      CALL QQPMH(Z,QPH,QMH)
```

```
      TERM2=TT*TT-T*T
      GTFCT=(TT*(DEL+TERM2)*QPH+T*(DEL-TERM2)*QMH)/
     *       (25.132741229*SQRT(TERM1*TERM1*TERM1)*Z*(Z+2.0))
      RETURN
      END


      SUBROUTINE GPLOT(K)
      REAL LAMBA
      DIMENSION XW(11),GMW(11)
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C PRINT AND PLOT ROUTINE FOR VORTEX DENSITY
      XW(1)=0.01
      DO 8 J=1,10
      IF(J-MM)4,4,6
    4 XW(J+1)=DELTA(J)
      GO TO 8
    6 XW(J+1)=-2.0**125
      GMW(J)=-2.0**125
    8 CONTINUE
      KGO=ITERT-ITBEG+1
      KGO=KGO-(NPLOT+1)*(KGO/(NPLOT+1))
      KGO=K-1+((KGO+2)*(KGO+3))/2
      KPEN=1
      DO 40 I=1,100
      XC=I/50.0
      TPRIM=TPFCT(K,XC)
      YC=GMFCT(2,K,XC)/SQRT(1.0+TPRIM*TPRIM)
      IF(0.25-ABS(YC))30,20,20
   20 XP=10-XC*5.0
      YP=YC*20.0+10.5
      CALL EPLOT(KPEN,YP,XP)
      KPEN=2
      IF(I-KGO*(I/KGO))40,30,40
   30 KPEN=1
   40 CONTINUE
      IF(K-1)50,50,60
   50 WRITE(3,3001)ITERT
   60 MPLUS=MM+1
      DO 80 I=1,MPLUS
      TPRIM=TPFCT(K,XW(I))
      GMW(I)=GMFCT(2,K,XW(I))/SQRT(1.0+TPRIM*TPRIM)
   80 CONTINUE
      WRITE(3,3002)K,(GMW(I),I=1,10),GMINF(2,K),(A(J),J=1,NN)
      IF(KK-K)100,100,90
   90 WRITE(3,3003)
  100 RETURN
 3001 FORMAT(11H ITERATION ,I2/2H '/2H ')
 3002 FORMAT(12H ' GAMMA SUB,I2,1X,11F9.5/2H '/11H ' A COEFFS,9X,5E18.10
     */(2H ',18X,5E18.10))
 3003 FORMAT(2H '/2H ')
      END


      SUBROUTINE INITL
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
```

```
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
      READ(2,2001)IM,ID,IY
      WRITE(3,3001)IM,ID,IY
      READ(2,2002)(KASE(I),I=1,5),LAMBA,NN,NP,MM,KK,ITBEG,ITEND,NPLOT,
     *                NSWTH
      WRITE(3,3002)(KASE(I),I=1,5),LAMBA,NN,NP,MM,KK,ITBEG,ITEND,NPLOT,
     *                 NSWTH
      KLES1=KK-1
      NPSI=2*KK+1
      IF(NPLOT)10,10,20
   10 NPLOT=1
   20 READ(2,2003)(RR(K),K=1,KK)
      WRITE(3,3003)(RR(K),K=1,KK)
      READ(2,2003)(CAPGM(K),K=1,KK)
      WRITE(3,3004)(CAPGM(K),K=1,KK)
      CAPGM(KK+1)=0.0
      READ(2,2003)(BETA(J),J=1,NN)
      WRITE(3,3005)(BETA(J),J=1,NN)
      IF(KLES1)40,40,30
   30 READ(2,2003)(ALPHA(J),J=1,NP)
      WRITE(3,3006)(ALPHA(J),J=1,NP)
   40 READ(2,2003)(DELTA(J),J=1,MM)
      WRITE(3,3007)(DELTA(J),J=1,MM)
      READ(2,2003)(C(J),J=1,NN)
      WRITE(3,3008)(C(J),J=1,NN)
      IF(ITBEG)130,100,50
   50 DO 60 K=1,KK
      READ(2,2004)(B(J),J=1,MM)
      WRITE(1'K)(B(J),J=1,MM)
   60 CONTINUE
      IF(KLES1)140,90,70
   70 DO 80 K=1,KLES1
      READ(2,2004)(A(J),J=1,NP)
      WRITE(2'K)(A(J),J=1,NP)
      K2=18+K
      WRITE(2'K2)(A(J),J=1,NP)
   80 CONTINUE
   90 READ(2,2004)(A(J),J=1,NN)
      WRITE(2'KK)(A(J),J=1,NN)
      K2=18+KK
      WRITE(2'K2)(A(J),J=1,NN)
      GO TO 140
  100 DO 110 J=1,10
      B(J)=0.0
  110 CONTINUE
      DO 115 J=1,11
      A(J)=0.0
  115 CONTINUE
      DO 120 K=1,KK
      WRITE(1'K)(B(J),J=1,10)
      WRITE(2'K)(A(J),J=1,11)
      K2=18+K
      WRITE(2'K2)(A(J),J=1,11)
  120 CONTINUE
      ITBEG=1
      GO TO 140
  130 ITBEG=-ITBEG+1
  140 DO 160 J=1,MM
      DO 150 NU=1,MM
      TRHS(J,NU)=FFCT(NU,DELTA(J))
```

```
  150 CONTINUE
  160 CONTINUE
      WRITE(3'1)((TRHS(J,NU),NU=1,MM),J=1,MM)
      IF(KLES1)200,200,170
  170 DO 190 J=1,NP
      DO 180 NU=1,NP
      TRHS(J,NU)=EXP(-NU*ALPHA(J))
  180 CONTINUE
  190 CONTINUE
      WRITE(4'1)((TRHS(J,NU),NU=1,NP),J=1,NP)
  200 RETURN
 2001 FORMAT(3I5)
 2002 FORMAT(5A2,F10.0,8I5)
 2003 FORMAT(8F10.0)
 2004 FORMAT(4E20.11)
 3001 FORMAT(1H1////' PRINTOUT OF INPUT FOR MAIN CODE'///1X,3I5//)
 3002 FORMAT(1X,5A2/10X,7HLAMBA =,F10.4/10X,4HNN =,I3/10X,4HNP =,I3/10X,
     *4HMM =,I3/10X,4HKK =,I3/10X,7HITBEG =,I3/10X,7HITEND =,I3/10X,
     *7HNPLOT =,I3/10X,7HNSWTH =,I3//)
 3003 FORMAT(//8H RR =    ,10F10.5/8X,8F10.5)
 3004 FORMAT(//8H CAPGM =,10F10.5/8X,8F10.5)
 3005 FORMAT(//8H BETA = ,11F10.5)
 3006 FORMAT(//8H ALPHA =,8F10.5)
 3007 FORMAT(//8H DELTA =,10F10.5)
 3008 FORMAT(//8H C =     ,11F10.5)
      END


      SUBROUTINE BPLOT
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
      WRITE(3,3001)IM,ID,IY,(KASE(I),I=1,5),ITERT
      IF(ITERT-ITBEG)10,50,20
   10 CALL SCALE(1.0,1.0,0.0,11.0)
      GO TO 40
   20 JPLOT=ITERT-ITBEG+1
      IF(JPLOT-NPLOT*(JPLOT/NPLOT))50,30,50
   30 CALL EPLOT(1,16.5,0.0)
      CALL SCALE(1.0,1.0,0.0,1.0)
   40 CALL EPLOT(1,0.0,0.0)
      CALL SCALE(1.0,1.0,0.0,-0.38)
      CALL EGRID(1,0.0,0.0,2.5,4)
      CALL EGRID(0,0.0,10.0,1.0,5)
      CALL EGRID(0,5.5,10.0,1.0,10)
      CALL EGRID(3,10.5,10.0,2.5,4)
   50 RETURN
 3001 FORMAT(1H1,I2,1H/,I2,1H/,I2,5X,5A2,5X,'ITERATION NUMBER',I3//)
      END


      FUNCTION TTFCT(K,X)
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C EQUATION 30
      IF(K-IREC1+1)10,20,10
```

```
   10 CALL GETCF(1,K,B,MM)
   20 Z=EXP(-X)
      BB=B(MM)
      TTFCT=B(1)
      DO 30 J=2,MM
      TTFCT=TTFCT+B(J)
      NU=MM-J+1
      BB=B(NU)+Z*BB
   30 CONTINUE
      TTFCT=RR(K)-TTFCT+Z*BB
      RETURN
      END


      FUNCTION TPFCT(K,X)
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
      IF(K-IREC1+1)10,20,10
C FIRST DERIVATIVE WITH RESPECT TO X OF EQUATION 30
   10 CALL GETCF(1,K,B,MM)
   20 Z=EXP(-X)
      CC=MM*B(MM)
      DO 30 J=2,MM
      NU=MM-J+1
      CC=NU*B(NU)+Z*CC
   30 CONTINUE
      TPFCT=-Z*CC
      RETURN
      END


      FUNCTION FFCT(J,X)
C EQUATION 32
      FFCT=EXP(-J*X)-1.0
      RETURN
      END


      FUNCTION GMFCT(NEWLD,K,X)
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C EQUATIONS 31 AND 33
      FAC=SQRT(1.0+TPFCT(K,X)**2)
      K2=18*(NEWLD-1)+K
      IF(KK-K)10,10,50
   10 IF(K2-IREC2+1)20,30,20
   20 CALL GETCF(2,K2,A,NN)
   30 GMFCT=0.0
      DO 40 J=1,NN
      GMFCT=GMFCT+A(J)*X**C(J)
   40 CONTINUE
      GMFCT=GMINF(NEWLD,K)*(1.0+EXP(-3.0*X)*GMFCT)*FAC
      RETURN
   50 IF(K2-IREC2+1)60,70,60
   60 CALL GETCF(2,K2,A,NP)
```

```
   70 Z=EXP(-X)
      EE=A(NP)
      DO 80 J=2,NP
      NU=NP-J+1
      EE=A(NU)+Z*EE
   80 CONTINUE
      GMFCT=GMINF(NEWLD,K)*(1.0+Z*EE)*FAC
      RETURN
      END


      SUBROUTINE GAMIN
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C EQUATION 29
      SUM=0.0
      DO 10 I=1,KK
      NU=KK-I+1
      TKSQR=TTFCT(NU,1000.0)**2
      FF=(CAPGM(NU)-CAPGM(NU+1)-(CAPGM(NU)**2-CAPGM(NU+1)**2)/
     *   (12.566308*TKSQR))/3.1415927
      GMINF(1,NU)=GMINF(2,NU)
      GMINF(2,NU)=-(LAMBA+SUM)+SQRT((LAMBA+SUM)**2+FF)
      SUM=SUM+GMINF(2,NU)
   10 CONTINUE
      RETURN
      END


      SUBROUTINE QQPMH(Z,QPH,QMH)
C LEGENDRE FUNCTIONS AS DESCRIBED IN APPENDIX A.2
      Y=Z/(Z+2.0)
      YLOG=ALOG(Y)
      SQROT=SQRT(Z+2.0)
      A=0.03742563713+0.01451196212*Y
      A=0.03590092383+A*Y
      A=0.09666344259+A*Y
      A=1.38629436112+A*Y
      B=0.03328355346+0.00441787012*Y
      B=0.06880248576+B*Y
      B=0.12498593597+B*Y
      B=0.5+B*Y
      C=0.04757383546+0.01736506451*Y
      C=0.06260601220+C*Y
      C=0.44325141463+C*Y
      C=1.0+C*Y
      D=0.04069697526+0.00526449639*Y
      D=0.09200180037+D*Y
      D=0.24998368310+D*Y
      D=D*Y
      QMH=1.414213562373/SQROT*(A-B*YLOG)
      QPH=(Z+1.0)*QMH-1.414213562373*SQROT*(C-D*YLOG)
      RETURN
      END
```

```
      SUBROUTINE ESOLV(A,B,SOLUT,N)
      DIMENSION A(11,11),B(11),SOLUT(11),IPIV(11),INDEX(11,2),PIVOT(11)
C SOLUTION OF EQUATIONS IN MATRIX FORM
      DETER=1.0
      DO 10 I=1,N
      IPIV(I)=0
   10 CONTINUE
      DO 200 I=1,N
      AMAX=0.0
      DO 60 J=1,N
      IF(IPIV(J)-1)20,60,20
   20 DO 50 K=1,N
      IF(IPIV(K)-1)30,50,210
   30 IF(ABS(AMAX)-ABS(A(J,K)))40,50,50
   40 IROW=J
      ICOL=K
      AMAX=A(J,K)
   50 CONTINUE
   60 CONTINUE
      IPIV(ICOL)=IPIV(ICOL)+1
      IF(IROW-ICOL)70,110,70
   70 DETER=-DETER
      DO 80 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOL,L)
      A(ICOL,L)=SWAP
   80 CONTINUE
      SWAP=B(IROW)
      B(IROW)=B(ICOL)
      B(ICOL)=SWAP
  110 INDEX(I,1)=IROW
      INDEX(I,2)=ICOL
      PIVOT(I)=A(ICOL,ICOL)
      DETER=DETER*PIVOT(I)
      A(ICOL,ICOL)=1.0
      DO 120 L=1,N
      A(ICOL,L)=A(ICOL,L)/PIVOT(I)
  120 CONTINUE
      B(ICOL)=B(ICOL)/PIVOT(I)
      DO 200 L1=1,N
      IF(L1-ICOL)160,200,160
  160 T=A(L1,ICOL)
      A(L1,ICOL)=0.0
      DO 170 L=1,N
      A(L1,L)=A(L1,L)-A(ICOL,L)*T
  170 CONTINUE
      B(L1)=B(L1)-B(ICOL)*T
  200 CONTINUE
  210 CONTINUE
      DO 220 I=1,N
      SOLUT(I)=B(I)
  220 CONTINUE
      RETURN
      END


      SUBROUTINE TAUFT(KGO,LL)
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
```

```
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C TAUS TO BE USED IN GAUSS-CHEBYSHEV AS DESCRIBED IN APPENDIX A.3
      IF(ITERT-NSWTH)10,20,20
   10 LL=10
      GO TO 30
   20 LL=10*2**(4-KGO)
   30 DO 70 L=1,LL
      TAU=COS(3.1415926536*(2.0*L-1.0)/(4.0*LL))
      GO TO (40,50,60), KGO
   40 TAU1(L)=TAU
      GO TO 70
   50 TAU2(L)=TAU
      GO TO 70
   60 TAU3(L)=TAU
   70 CONTINUE
      RETURN
      END


      SUBROUTINE GETCF(NFILE,NRECD,VECTR,NUMBR)
      DIMENSION VECTR(1)
C RETRIEVE CORRECT SET OF A OR B COEFFICIENTS
      READ(NFILE'NRECD)(VECTR(I),I=1,NUMBR)
      RETURN
      END


      SUBROUTINE PUTCF
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *ITBEG,ITEND,NPLOT,NSWTH,NPSI,ITERT,PSINF(18),BETA(11),ALPHA(8),
     *DELTA(10),TRHS(11,11),TLHS(11),TAU1(80),TAU2(40),TAU3(20)
C STORE A & B COEFFICIENTS TO BE PUNCHED AND USED AS INPUT
      DO 40 K=1,KK
      IREC=18*(ITERT-1)+K
      CALL GETCF(1,K,B,MM)
      WRITE(6'IREC)(B(J),J=1,MM)
   10 IF(KK-K)20,20,30
   20 CALL GETCF(2,K,A,NN)
      WRITE(7'IREC)(A(J),J=1,NN)
      GO TO 40
   30 CALL GETCF(2,K,A,NP)
      WRITE(7'IREC)(A(J),J=1,NP)
   40 CONTINUE
      RETURN
      END
```

```
C     S T R E A M T U B E   C O D E
C
C     P S N E T
C
C ALL INPUT FOR THE STREAMLINE CODE IS DONE IN SUBROUTINE INPUT.
C INPUT VARIABLES ARE AS FOLLOWS -
C
C VARIABLE NAME      DESCRIPTION
C -------------      -----------------------------------------------------------
C            IM      TWO DIGIT MONTH OF THE YEAR
C            ID      TWO DIGIT DAY OF THE MONTH
C            IY      TWO DIGIT YEAR
C          KASE      TEN CHARACTER CASE IDENTIFICATION
C         LAMBA      ADVANCE RATIO
C            NN      NUMBER OF TERMS IN VORTEX DENSITY FUNCTION FOR OUTER
C                         VORTEX TUBE
C            NP      NUMBER OF TERMS IN VORTEX DENSITY FUNCTION FOR INNER
C                         VORTEX TUBES
C            MM      NUMBER OF TERMS IN STREAM FUNCTION
C            KK      NUMBER OF STEPS IN THE PIECEWISE CONSTANT CIRCULATION
C                         DISTRIBUTION
C           NPF      NUMBER OF FIELD POINTS
C           NX1      NUMBER OF AXIAL MESH POINTS IN REGION 1
C           NR1      NUMBER OF RADIAL MESH POINTS IN REGION 1
C           NX2      NUMBER OF AXIAL MESH POINTS IN REGION 2
C           NR2      NUMBER OF RADIAL MESH POINTS IN REGION 2
C            RR      RADIAL LOCATIONS OF STEPS IN CIRCULATION DISTRIBUTION
C         CAPGM      PIECEWISE CONSTANT CIRCULATION DISTRIBUTION
C             B      COEFFICIENTS IN REPRESENTATION OF VORTEX TUBE SHAPE
C             A      COEFFICIENTS IN REPRESENTATION OF VORTEX DENSITY
C             C      EXPONENTS IN MATCHING FUNCTION FOR VORTEX DENSITY ON
C                         OUTER VORTEX TUBE
C            XF      AXIAL POSITIONS OF FIELD POINTS
C            RF      RADIAL POSITIONS OF FIELD POINTS
C            X1      AXIAL POSITIONS OF MESH POINTS IN REGION 1
C            X2      AXIAL POSITIONS OF MESH POINTS IN REGION 2
C            R1      RADIAL POSITIONS OF MESH POINTS IN REGION 1
C             C      UPPER BOUND FOR RADIAL MESH POINTS IN REGION 2
C
C NOTES ON INPUT -
C 1   IF KK=1, NP NEED NOT BE SPECIFIED
C
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *NPF,NX1,NR1,NX2,NR2,XF(25),RF(25),X1(20),R1(25),X2(27),R2(20),
     *PSI(25),TAU1(80),TAU2(40)
      DEFINE FILE 1(18,32,U,IREC1), 2(36,33,U,IREC2), 3(100,75,U,IREC3)
      CALL INPUT(CC)
      CALL GAMIN
      IREC=1
      WRITE(3,3001)
      DO 5 I=1,NPF
      IF(ABS(XF(I))-0.05)2,3,3
    2 CALL PINTG(TAU1,80,XF(I),RF(I),1,KK,PSI,LAMBA)
      GO TO 4
    3 CALL PINTG(TAU2,40,XF(I),RF(I),1,KK,PSI,LAMBA)
    4 CALL OUTPT(IREC,XF(I),RF(I),1,PSI)
    5 CONTINUE
      WRITE(3,3002)
```

```
      DO 10 I=1,NX1
      IF(ABS(X1(I))-0.05)6,7,7
    6 CALL PINTG(TAU1,80,X1(I),R1,NR1,KK,PSI,LAMBA)
      GO TO 8
    7 CALL PINTG(TAU2,40,X1(I),R1,NR1,KK,PSI,LAMBA)
    8 CALL OUTPT(IREC,X1(I),R1,NR1,PSI)
   10 CONTINUE
      WRITE(3,3003)
      DO 30 I=1,NX2
      TT=TTFCT(KK,X2(I))
      DO 20 J=1,NR2
      R2(J)=TT+(CC-TT)*(J-1.0)/(NR2-1.0)
   20 CONTINUE
      IF(ABS(X2(I))-0.05)14,16,16
   14 CALL PINTG(TAU1,80,X2(I),R2,NR2,KK,PSI,LAMBA)
      GO TO 18
   16 CALL PINTG(TAU2,40,X2(I),R2,NR2,KK,PSI,LAMBA)
   18 CALL OUTPT(IREC,X2(I),R2,NR2,PSI)
   30 CONTINUE
      WRITE(3,3004)
      DO 40 I=1,NPF
      READ(3'I)PSI(I)
      WRITE(3,3005)PSI(I)
      IREC=NPF
      CALL REGN1(IREC,NX1,NR1,X1,R1,PSI(I))
      CALL REGN2(IREC,NX2,NR2,X2,R2,PSI(I),KK,CC)
   40 CONTINUE
      CALL EXIT
 3001 FORMAT('1FIELD POINTS'/)
 3002 FORMAT(////' REGION 1 POINTS'/)
 3003 FORMAT(/////' REGION 2 POINTS'/)
 3004 FORMAT(1H1)
 3005 FORMAT(////' PSI=',E17.10,' IS AT')
      END


      SUBROUTINE INPUT(CC)
      REAL LAMBA
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11),
     *NPF,NX1,NR1,NX2,NR2,XF(25),RF(25),X1(20),R1(25),X2(27),R2(20),
     *PSI(25),TAU1(80),TAU2(40)
      READ(2,2001)IM,ID,IY
      WRITE(3,3001)IM,ID,IY
      READ(2,2002)(KASE(I),I=1,5),LAMBA,NN,NP,MM,KK,NPF,NX1,NR1,NX2,NR2
      WRITE(3,3002)(KASE(I),I=1,5),LAMBA,NN,NP,MM,KK,NPF,NX1,NR1,NX2,NR2
      READ(2,2003)(RR(K),K=1,KK)
      WRITE(3,3003)(RR(K),K=1,KK)
      READ(2,2003)(CAPGM(K),K=1,KK)
      WRITE(3,3004)(CAPGM(K),K=1,KK)
      CAPGM(KK+1)=0.0
      WRITE(3,3005)
      DO 10 K=1,KK
      READ(2,2004)(B(J),J=1,MM)
      WRITE(3,3006)(B(J),J=1,MM)
      WRITE(1'K)(B(J),J=1,MM)
   10 CONTINUE
      WRITE(3,3007)
      DO 50 K=1,KK
      K2=18+K
   20 IF(KK-K)30,30,40
```

```
   30 READ(2,2004)(A(J),J=1,NN)
      WRITE(3,3006)(A(J),J=1,NN)
      WRITE(2'K2)(A(J),J=1,NN)
      GO TO 50
   40 READ(2,2004)(A(J),J=1,NP)
      WRITE(3,3006)(A(J),J=1,NP)
      WRITE(2'K2)(A(J),J=1,NP)
   50 CONTINUE
      READ(2,2003)(C(J),J=1,NN)
      WRITE(3,3008)(C(J),J=1,NN)
      READ(2,2003)(XF(I),I=1,NPF)
      WRITE(3,3009)(XF(I),I=1,NPF)
      READ(2,2003)(RF(I),I=1,NPF)
      WRITE(3,3010)(RF(I),I=1,NPF)
      READ(2,2003)(X1(I),I=1,NX1)
      WRITE(3,3011)(X1(I),I=1,NX1)
      READ(2,2003)(R1(I),I=1,NR1)
      WRITE(3,3012)(R1(I),I=1,NR1)
      READ(2,2003)(X2(I),I=1,NX2)
      WRITE(3,3013)(X2(I),I=1,NX2)
      READ(2,2003)CC
      WRITE(3,3014)CC
      DO 60 L=1,80
      TAU1(L)=COS((2.0*L-1.0)*3.1415926536/320.0)
   60 CONTINUE
      DO 70 L=1,40
      TAU2(L)=COS((2.0*L-1.0)*3.1415926536/160.0)
   70 CONTINUE
      RETURN
 2001 FORMAT(3I5)
 2002 FORMAT(5A2,F10.0,9I5)
 2003 FORMAT(8F10.0)
 2004 FORMAT(4E20.10)
 3001 FORMAT(1H1////' PRINTOUT OF INPUT FOR STREAMLINE CODE'///1X,3I5//)
 3002 FORMAT(1X,5A2/10X,7HLAMBA =,F10.4/10X,4HNN =,I3/10X,4HNP =,I3/10X,
     *4HMM =,I3/10X,4HKK =,I3/10X,5HNPF =,I3/10X,5HNX1 =,I3/10X,5HNR1 =,
     *I3/10X,5HNX2 =,I3/10X,5HNR2 =,I3//)
 3003 FORMAT(//8H RR =    ,10F10.5/8X,8F10.5)
 3004 FORMAT(//8H CAPGM =,10F10.5/8X,8F10.5)
 3005 FORMAT(////' SHAPE COEFFICIENTS'/)
 3006 FORMAT(6E20.10)
 3007 FORMAT(////' GAMMA COEFFICIENTS'/)
 3008 FORMAT(//' C (EXPONENTS FOR G FUNCTION WITH K=KK) ='/11F10.5)
 3009 FORMAT(////' FIELD POINTS, X='(/10F12.6))
 3010 FORMAT(/    ' FIELD POINTS, R='(/10F12.6))
 3011 FORMAT(////' REGION 1 MESH POINTS, X='(/10F12.6))
 3012 FORMAT(/    ' REGION 1 MESH POINTS, R='(/10F12.6))
 3013 FORMAT(////' REGION 2 MESH POINTS, X='(/10F12.6))
 3014 FORMAT(/    ' REGION 2 UPPER BOUND, C='/F12.6)
      END


      SUBROUTINE PINTG(TAU,LL,X,R,NR,KK,PSI,LAMBA)
      REAL LAMBA
      DIMENSION TAU(1),R(1),PSI(1)
      DO 10 I=1,NR
      PSI(I)=0.0
   10 CONTINUE
      DO 70 K=1,KK
      DO 60 L=1,LL
      IF(X-0.00001)20,20,40
```

```
 20 S1=2.5*(1.0+TAU(L))
    S2=2.5*(1.0-TAU(L))
    X1=S1**1.1+S1*S1*S1
    X2=S2**1.1+S2*S2*S2
    T1=TTFCT(K,X1)
    T2=TTFCT(K,X2)
    GAM1=GMFCT(2,K,X1)
    GAM2=GMFCT(2,K,X2)
    ROOT=SQRT(1.0-TAU(L)**2)
    TERM1=ROOT*(1.1*S1**0.1+3.0*S1*S1)*GAM1
    TERM2=ROOT*(1.1*S2**0.1+3.0*S2*S2)*GAM2
    DO 30 I=1,NR
    PSI(I)=PSI(I)+TERM1*GGFCT(X1,T1,X,R(I))+TERM2*GGFCT(X2,T2,X,R(I))
 30 CONTINUE
    GO TO 60
 40 S3=2.5*(1.0+TAU(L))
    S4=2.5*(1.0-TAU(L))
    X1=X*(1.0+TAU(L))/2.0
    X2=X*(1.0-TAU(L))/2.0
    X3=X+S3+S3*S3*S3
    X4=X+S4+S4*S4*S4
    T1=TTFCT(K,X1)
    T2=TTFCT(K,X2)
    T3=TTFCT(K,X3)
    T4=TTFCT(K,X4)
    GAM1=GMFCT(2,K,X1)
    GAM2=GMFCT(2,K,X2)
    GAM3=GMFCT(2,K,X3)
    GAM4=GMFCT(2,K,X4)
    ROOT=SQRT(1.0-TAU(L)**2)
    TERM1=ROOT*0.2*X*GAM1
    TERM2=ROOT*0.2*X*GAM2
    TERM3=ROOT*(1.0+3.0*S3*S3)*GAM3
    TERM4=ROOT*(1.0+3.0*S4*S4)*GAM4
    DO 50 I=1,NR
    PSI(I)=PSI(I)+TERM1*GGFCT(X1,T1,X,R(I))+TERM2*GGFCT(X2,T2,X,R(I))+
   *                TERM3*GGFCT(X3,T3,X,R(I))+TERM4*GGFCT(X4,T4,X,R(I))
 50 CONTINUE
 60 CONTINUE
 70 CONTINUE
    CONST=2.5*3.1415926536/(2.0*LL)
    DO 100 I=1,NR
    IF(ABS(R(I))-0.00001)80,80,90
 80 PSI(I)=0.0
    GO TO 100
 90 PSI(I)=LAMBA*R(I)**2/2.0+CONST*PSI(I)
100 CONTINUE
    RETURN
    END


    SUBROUTINE OUTPT(IREC,X,R,NR,PSI)
    DIMENSION R(1),PSI(1)
    DATA KXEQL/'X='/, KREQL/'R='/
    J1=1
    DO 40 I=1,NR,10
    IF(J1+9-NR)10,20,20
 10 J2=J1+9
    GO TO 30
 20 J2=NR
 30 WRITE(3,3001)(KXEQL,X,J=J1,J2)
```

```
      WRITE(3,3002)(KREQL,R(J),J=J1,J2)
      WRITE(3,3003)(PSI(J),J=J1,J2)
      J1=J2+1
   40 CONTINUE
      WRITE(3'IREC)(PSI(I),I=1,NR)
      IREC=IREC+1
      RETURN
 3001 FORMAT(/10(2X,A2,F7.4,1X))
 3002 FORMAT(10(2X,A2,F7.4,1X))
 3003 FORMAT(10E12.5)
      END


      SUBROUTINE REGN1(IREC,NX1,NR1,X1,R1,PSI)
      DIMENSION X1(20),R1(20),P(20,20)
      DO 10 IX=1,NX1
      IREC=IREC+1
      READ(3'IREC)(P(IX,IR),IR=1,NR1)
   10 CONTINUE
      IF(NX1-1)100,100,15
   15 WRITE(3,3001)
      DO 40 IR=1,NR1
      DO 30 IX=3,NX1
      CALL BETWN(P(IX,IR),P(IX-1,IR),PSI,KGO)
      GO TO (20,30), KGO
   20 CALL THRPT(X1(IX-2),X1(IX-1),X1(IX),P(IX-2,IR),P(IX-1,IR),P(IX,IR)
     *          ,PSI,X)
      WRITE(3,3002)X,R1(IR)
   30 CONTINUE
   40 CONTINUE
      WRITE(3,3003)
      DO 90 IX=1,NX1
      CALL BETWN(P(IX,2),P(IX,1),PSI,KGO)
      GO TO (50,60), KGO
   50 CALL THRPT(-R1(1),R1(1),R1(2),P(IX,1),P(IX,1),P(IX,2),PSI,R)
      WRITE(3,3002)X1(IX),R
   60 DO 80 IR=3,NR1
      CALL BETWN(P(IX,IR),P(IX,IR-1),PSI,KGO)
      GO TO (70,80), KGO
   70 CALL THRPT(R1(IR-2),R1(IR-1),R1(IR),P(IX,IR-2),P(IX,IR-1),P(IX,IR)
     *          ,PSI,R)
      WRITE(3,3002)X1(IX),R
   80 CONTINUE
   90 CONTINUE
  100 RETURN
 3001 FORMAT(30X,'REGION 1, AXIAL SCAN')
 3002 FORMAT(3X,2HX=,F7.4,5X,2HR=,F7.4)
 3003 FORMAT(30X,'REGION 1, RADIAL SCAN')
      END


      SUBROUTINE REGN2(IREC,NX2,NR2,X2,R2,PSI,KK,CC)
      DIMENSION X2(27),R2(20),P(27,20)
      DO 10 IX=1,NX2
      IREC=IREC+1
      READ(3'IREC)(P(IX,IR),IR=1,NR2)
   10 CONTINUE
      IF(NX2-1)60,60,15
   15 NRSCN=NR2-1
      WRITE(3,3001)
      DO 50 IX=1,NX2
```

```
      TT=TTFCT(KK,X2(IX))
      DO 20 IR=1,NR2
      R2(IR)=TT+(CC-TT)*(IR-1.0)/(NR2-1.0)
   20 CONTINUE
      DO 40 IR=2,NRSCN
      CALL BETWN(P(IX,IR),P(IX,IR-1),PSI,KGO)
      GO TO (30,40), KGO
   30 CALL THRPT(R2(IR+1),R2(IR),R2(IR-1),P(IX,IR+1),P(IX,IR),P(IX,IR-1)
     *          ,PSI,R)
      WRITE(3,3002)X2(IX),R
   40 CONTINUE
   50 CONTINUE
   60 RETURN
 3001 FORMAT(30X,'REGION 2, RADIAL SCAN')
 3002 FORMAT(3X,2HX=,F7.4,5X,2HR=,F7.4)
      END


      SUBROUTINE THRPT(T1,T2,T3,P1,P2,P3,PSI,V)
      CCOEF=(P3-P1+(T1-T3)*(P1-P2)/(T1-T2))/ (T1*T2-T1*T3-T2*T3+T3*T3)
      BCOEF=(P1-P2)/(T1-T2)-CCOEF*(T1+T2)
      ACOEF=P1-BCOEF*T1-CCOEF*T1*T1
      V=SQRT(BCOEF*BCOEF-4.0*CCOEF*(ACOEF-PSI))
      VPLUS=-(BCOEF-V)/(2.0*CCOEF)
      VMINS=-(BCOEF+V)/(2.0*CCOEF)
      CALL BETWN(T2,T3,VPLUS,KGO)
      GO TO (10,20), KGO
   10 V=VPLUS
      RETURN
   20 V=VMINS
      RETURN
      END


      SUBROUTINE BETWN(P1,P2,P,KGO)
      D12=ABS(P2-P1)
      D1P=ABS(P1-P)
      D2P=ABS(P2-P)
      IF(ABS(D2P+D1P-D12)-0.00000001)10,10,20
   10 KGO=1
      RETURN
   20 KGO=2
      RETURN
      END
```

```
C     V E L O C I T Y   B O X   C O D E
C
C     V F E L D
C
C ALL INPUT FOR THE VELOCITY BOX CODE IS DONE IN THE MAINLINE PROGRAM.
C INPUT VARIABLES ARE AS FOLLOWS -
C
C VARIABLE NAME      DESCRIPTION
C -------------      ------------------------------------------------------------
C            IM      TWO DIGIT MONTH OF THE YEAR
C            ID      TWO DIGIT DAY OF THE MONTH
C            IY      TWO DIGIT YEAR
C          KASE      TEN CHARACTER CASE IDENTIFICATION
C         LAMBA      ADVANCE RATIO
C            NN      NUMBER OF TERMS IN VORTEX DENSITY FUNCTION FOR OUTER
C                         VORTEX TUBE
C            NP      NUMBER OF TERMS IN VORTEX DENSITY FUNCTION FOR INNER
C                         VORTEX TUBES
C            MM      NUMBER OF TERMS IN STREAM FUNCTION
C            KK      NUMBER OF STEPS IN THE PIECEWISE CONSTANT CIRCULATION
C                         DISTRIBUTION
C            RR      RADIAL LOCATIONS OF STEPS IN CIRCULATION DISTRIBUTION
C         CAPGM      PIECEWISE CONSTANT CIRCULATION DISTRIBUTION
C             B      COEFFICIENTS IN REPRESENTATION OF VORTEX TUBE SHAPE
C             A      COEFFICIENTS IN REPRESENTATION OF VORTEX DENSITY
C             C      EXPONENTS IN MATCHING FUNCTION FOR VORTEX DENSITY ON
C                         OUTER VORTEX TUBE
C             X      AXIAL POSITION FOR CALCULATING VELOCITIES
C             R      RADIAL POSITION FOR CALCULATING VELOCITIES
C
C NOTES ON INPUT -
C 1   IF KK=1, NP NEED NOT BE SPECIFIED
C
      REAL LAMBA
      DIMENSION XI(130),WT(130),XLIM(14),GU(5),GR(5)
      COMMON IM,ID,IY,KASE(5),LAMBA,NN,NP,MM,KK,IREC1,IREC2,
     *RR(18),CAPGM(19),GMINF(2,18),A(11),B(10),C(11)
      DATA XLIM/0.000,0.001,0.010,0.100,0.500,1.100,1.190,1.200,1.210,
     *          1.300,2.000,10.000,50.000,200.000/
      DATA GU/0.0744371695,0.2166976971,0.3397047841,0.4325316866,
     *        0.4869532643/
      DATA GR/0.1477621124,0.1346333597,0.1095431813,0.07472567458,
     *        0.03333567215/
      DEFINE FILE 1(18,32,U,IREC1), 2(36,33,U,IREC2)
      READ(2,2001)IM,ID,IY
      WRITE(3,3001)IM,ID,IY
      READ(2,2002)(KASE(I),I=1,5),LAMBA,NN,NP,MM,KK
      WRITE(3,3002)(KASE(I),I=1,5),LAMBA,NN,NP,MM,KK
      READ(2,2003)(RR(K),K=1,KK)
      WRITE(3,3003)(RR(K),K=1,KK)
      READ(2,2003)(CAPGM(K),K=1,KK)
      WRITE(3,3004)(CAPGM(K),K=1,KK)
      CAPGM(KK+1)=0.0
      WRITE(3,3005)
      DO 10 K=1,KK
      READ(2,2004)(B(J),J=1,MM)
      WRITE(3,3006)(B(J),J=1,MM)
      WRITE(1'K)(B(J),J=1,MM)
   10 CONTINUE
      WRITE(3,3007)
```

```
      DO 50 K=1,KK
      K2=18+K
   20 IF(KK-K)30,30,40
   30 READ(2,2004)(A(J),J=1,NN)
      WRITE(3,3006)(A(J),J=1,NN)
      WRITE(2'K2)(A(J),J=1,NN)
      GO TO 50
   40 READ(2,2004)(A(J),J=1,NP)
      WRITE(3,3006)(A(J),J=1,NP)
      WRITE(2'K2)(A(J),J=1,NP)
   50 CONTINUE
      READ(2,2003)(C(J),J=1,NN)
      WRITE(3,3008)(C(J),J=1,NN)
      DO 60 L=1,13
      D=XLIM(L+1)-XLIM(L)
      S=(XLIM(L+1)+XLIM(L))/2.0
      DO 58 I=1,10
      ISUB=10*(L-1)+I
      IF(I-5)54,54,56
   54 XI(ISUB)=S+D*GU(I)
      WT(ISUB)=D*GR(I)
      GO TO 58
   56 IELV=11-I
      XI(ISUB)=S-D*GU(IELV)
      WT(ISUB)=D*GR(IELV)
   58 CONTINUE
   60 CONTINUE
      CALL GAMIN
      WRITE(3,3009)
   70 READ(2,2005)X,R
      IF(R)999,80,80
   80 CALL UVINT(XI,WT,130,LAMBA,KK,X,R,U,V,ZETA,THETA)
      IF(R)999,90,100
   90 WRITE(3,3010)X,R,U
      GO TO 70
  100 WRITE(3,3011)X,R,U,V,ZETA,THETA
      GO TO 70
  999 CALL EXIT
 2001 FORMAT(3I5)
 2002 FORMAT(5A2,F10.0,4I5)
 2003 FORMAT(8F10.0)
 2004 FORMAT(4E20.10)
 2005 FORMAT(2F10.0)
 3001 FORMAT(1H1////' PRINTOUT OF INPUT FOR VELOCITY BOX CODE'///1X,3I5/
     */)
 3002 FORMAT(1X,5A2/10X,7HLAMBA =,F10.4/10X,4HNN =,I3/10X,4HNP =,I3/10X,
     *4HMM =,I3/10X,/)
 3003 FORMAT(//8H RR =    ,10F10.5/8X,8F10.5)
 3004 FORMAT(//8H CAPGM =,10F10.5/8X,8F10.5)
 3005 FORMAT(////' SHAPE COEFFICIENTS'/)
 3006 FORMAT(6E20.10)
 3007 FORMAT(////' GAMMA COEFFICIENTS'/)
 3008 FORMAT(//' C (EXPONENTS FOR G FUNCTION WITH K=KK) ='/11F10.5)
 3009 FORMAT(1H1,18X,1HX,14X,1HR,24X,1HU,19X,1HV,16X,4HZETA,15X,5HTHETA)
 3010 FORMAT(F20.5,F15.5,E25.10)
 3011 FORMAT(F20.5,F15.5,E25.10,3E20.10)
      END
```

```
      SUBROUTINE UVINT(XI,WT,LL,LAMBA,KK,X,R,U,V,ZETA,THETA)
      REAL LAMBA
      DIMENSION XI(1),WT(1)
      U=0.0
      V=0.0
      DO 40 K=1,KK
      DO 30 L=1,LL
      GAM=GMFCT(2,K,XI(L))
      DEL=(X-XI(L))**2
      T=TTFCT(K,XI(L))
      IF(R)10,20,10
   10 TTT=R*T
      Z=((R-T)**2+DEL)/(2.0*TTT)
      CALL QQPMH(Z,QPH,QMH)
      U=U+WT(L)*(R*(DEL+R*R-T*T)*QPH+T*(DEL+T*T-R*R)*QMH)
     *          /(25.132741229*SQRT(TTT*TTT*TTT)*Z*(Z+2.0))*GAM
      V=V+WT(L)*(X-XI(L))*((1.0+Z)*QPH-QMH)/(12.5663706*SQRT(TTT)
     *           *Z*(Z+2.0))*GAM
      GO TO 30
   20 U=U+WT(L)*T*T/SQRT((DEL+T*T)**3)*GAM
   30 CONTINUE
   40 CONTINUE
      IF(R)50,100,50
   50 U=LAMBA+U/R
      V=-V/R
      ZETA=SQRT(U*U+V*V)
      THETA=57.29578*ATAN(V/U)
      IF(U)60,90,90
   60 IF(V)70,80,80
   70 THETA=THETA-180.0
      RETURN
   80 THETA=THETA+180.0
   90 RETURN
  100 U=LAMBA+U/2.0
      RETURN
      END
```

REFERENCES

1. Aerodynamic Problems Associated With V/STOL Aircraft, CAL/USAAVLABS Symposium Proceedings, 22-24 June 1966, Buffalo, N. Y.

2. Erickson, J. C., Jr., A Continuous Vortex Sheet Representation Of Deformed Wakes Of Hovering Propellers, Proceedings of Third CAL/AVLABS Symposium on Aerodynamics of Rotary Wing and V/STOL Aircraft, Vol. 1, 18-20 June 1969, Buffalo, N. Y.

3. Hough, G. R., and Ordway, D. E., The Generalized Actuator Disk, Developments in Theoretical and Applied Mechanics, Vol. II, Pergamon Press, 1965.

4. Wu, T. Y., Flow Through A Heavily Loaded Actuator Disk, Schiffstechnik, Vol. 9, No. 47, pp. 134-138, 1962.

5. Cox, B. W., Vortex Ring Solutions Of Axisymmetric Propeller Flow Problems, Massachusetts Institute of Technology, Department of Naval Architecture and Marine Engineering Report No. 68-13, June 1968.

6. Greenberg, M. D., and Kaskel, A. L., Inviscid Flow Field Induced By A Rotor In Ground Effect, NASA CR-1027, May 1968.

7. Heyson, H. H., and Katzoff, S., Induced Velocities Near A Lifting Rotor With Nonuniform Disk Loading, NACA Report 1319, 1957.

8. Chaplin, H. R., A Method For Numerical Calculation Of Slipstream Contraction Of A Shrouded Impulse Disc In The Static Case With Application To Other Axisymmetric Potential Flow Problems, DTMB Aero Report 1077, June 1964.

9. Hough, G. R., and Ordway, D. E., Mean Flow Streamlines Of A Finite-Bladed Propeller, Journal of Aircraft, Vol. 4, No. 6, pp. 561-562, 1967.

10. Rorke, J. B., and Wells, C. D., A Hover Performance Analysis Combining The Strip-Momentum And Prescribed Wake Theories, Proceedings of Third CAL/AVLABS Symposium on Aerodynamics of Rotary Wing and V/STOL Aircraft, Vol. 1, 18-20 June 1969, Buffalo, N. Y.

11. Hastings, C., Jr., Approximations For Digital Computers, Princeton University Press, 1955.

12. Abramowitz, M., and Stegun, I., Handbook Of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series, No. 55, 1964.

NASA-Langley, 1970 — 1 CR-1672