JOURNAL OF COMPUTATIONAL PHYSICS 85, 104–129 (1989)

An Open Boundary Condition for the Computation of the Steady Incompressible Navier–Stokes Equations

F. NATAF

Centre de Mathématiques Appliquées de l'Ecole Polytechnique, 91128, Palaiseau Cedex, France

Received May 20, 1988

We design a new boundary condition on the stream function at outflow for the steady incompressible Navier–Stokes equations written in the stream function–vorticity formulation by using a Fourier transform in the direction perpendicular to the flow. Our open boundary condition is compared to the Dirichlet and Neumann boundary conditions by computing the flow past an ellipse of excentricity 50%. Our boundary condition proves to be the most accurate and thus the most time saving.

1. INTRODUCTION

Many numerical simulations of viscous fluid flow problems in physically unbounded exterior domains are carried out in "truncated" physical computational domains. Outflow boundary conditions such as Neumann or Dirichlet boundary conditions for the dependent variables are then usually prescribed. This careless truncation leads sometimes to instability in the numerical scheme and very often to inaccuracy in the solution. The only way to improve the accuracy is to use a quite large computational domain. The cost of the simulation is then increased. Thus, the accuracy of the outflow boundary condition and the computational cost are closely related and a more costly boundary condition that can be used closer to the body can in fact save computational time if the simulation is considered as a whole. The effects of truncation have been considered by Moretti [1], Moretti and Abbet [2], Wang and Longwell [3], and by Dekruif and Hassan [4].

The ideal would be to use outflow boundary conditions such that the solution in the truncated domain is equal to the solution in the infinite domain. Such boundary conditions will be referred to as exact open boundary conditions. Such boundary conditions for the linearized Navier–Stokes equations (called Oseen equations and which are also valid at high Reynolds number, see Vorus [5]) in the velocity–pressure formulation and for a domain unbounded in the direction perpendicular to the flow have been designed by Halpern [6] and Halpern and Schatzmann [7]. These boundary conditions are then applied to the Navier–Stokes equations. Due

0021-9991/89 $3.00

to the nonlinearity, they are no longer exact and will be referred to as open boundary conditions. The purpose of this paper is to design open boundary conditions for the steady state in the stream function–vorticity formulation in the case when the domain is bounded in the direction perpendicular to the flow. The steady state is computed as the limit in time of the unsteady Navier–Stokes equations. The vorticity is the solution to a transport equation for which local open boundary conditions have been proposed in [6] and are used here. Thus, only the new boundary condition on the stream function is tested. Taking advantage of the fact that we want to compute the steady state, the Poisson equation for the stream function is solved by coupling with the steady problem in the exterior domain and using eigenmode expansions. Coupling methods with simpler problems in the exterior domain are widely used, especially in fluid mechanics in connection with integral methods (see Johnson and Nedelec [8], Sequeria [9]) or Fourier methods (see Ferm [10], Ferm and Gustafsson [11], Lenoir [12], Hagström [13]). Let us notice that Fornberg [14] designed an open boundary condition in the case where the vorticity can be neglected by using explicit solutions of flow around vortex rings. It seems that our outflow boundary condition and Fornberg's are quite similar when the vorticity can be neglected at the boundary.

The paper is organized as follows: in Section 2, we present Navier–Stokes equations in stream function–vorticity formulation. In Section 3, we present the numerical method used for solving the Navier–Stokes equations. In Section 4, we design a new boundary condition at outflow for the stream function. In Section 5, we present the numerical implementation of our boundary condition. In Section 6, the results obtained with the boundary conditions designed in Section 4 are compared to the ones obtained with Dirichlet and Neumann boundary conditions. In Section 7, we conclude.

2. Navier–Stokes Equations

Throughout this paper, we consider the 2-dimensional unsteady incompressible Navier–Stokes equations. The velocity–pressure formulation is:

$$\begin{aligned}
&\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)\\
&\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)\quad (x,\,y)\in\mathbb{R}^2/\Omega_c,\ t\in[0,\,\infty[\qquad (1.1)\\
&\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0,
\end{aligned}$$

where t is the time, u and v are the components of the velocity in the x and y coordinate directions, ρ is the density, p is the pressure, and ν is the kinematic viscosity, Ω_c is the region occupied by the body (see Fig. 1). $\partial\Omega_c$ is the boundary of Ω_c.

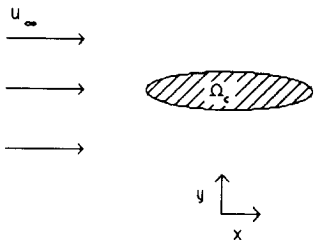


FIG. 1. Physical domain.

The stream function vorticity formulation is obtained by introducing ψ the stream function defined by

$$\frac{\partial \psi}{\partial y} = u \qquad \text{and} \qquad \frac{\partial \psi}{\partial x} = -v. \tag{1.2}$$

The existence of ψ is due to the incompressibility of the fluid. The vorticity ω is:

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \tag{1.3}$$

The unsteady Navier–Stokes equations are written in dimensionless form in the stream function–vorlticity formulation:

$$\begin{aligned} &\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \frac{2}{\mathrm{Re}}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) \\ &\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \end{aligned} \tag{1.4}$$

(Re is a Reynolds number). As for the boundary conditions, they consist of boundary conditions on the body and at infinity. The no-slip condition on $\partial\Omega_c$ is $\psi_x = 0$ and $\psi_y = 0$. These two boundary conditions on ψ have to induce one condition on ψ and one on ω. This is done in a classical way: at the rigid wall, the tangential velocity is zero and, if n denotes the exterior normal at the wall, $\partial^2\psi/\partial n^2 = -\omega$. The boundary conditions at infinity are easy to handle. Indeed, v tends to zero and u to u_∞ at infinity.

Thus,

$$|\omega \to 0 \tag{1.5a}$$

$$|\psi \sim u_\infty \times y + c^{st} \qquad \text{as} \quad |x| \to \infty. \tag{1.5b}$$

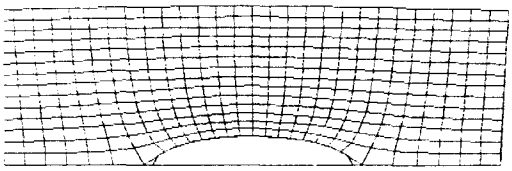


FIG. 2. Physical grid near the ellipse.

3. OUTLINES OF THE NUMERICAL CODE

This code computes the fluid flow past an ellipse of variable excentricity. The ellipse is aligned with the stream and the flow is thus symmetrical. The grid is made up with streamlines and equipotential lines for an analytic solution to the incompressible Euler equations. This grid is orthogonal and matches the streamlines of the perfect fluid flow (see Fig. 2). By use of a conformal map, this grid is mapped to a rectangular grid (see Fig. 3) and the ellipse to a segment on the axis of symmetry. The transformed Navier–Stokes equations (1.4) take the form

$$f\frac{\partial\omega}{\partial t}+u\frac{\partial\omega}{\partial X}+v\frac{\partial\omega}{\partial Y}=\frac{2}{\mathrm{Re}}\left(\frac{\partial^2\omega}{\partial X^2}+\frac{\partial^2\omega}{\partial Y^2}\right) \tag{2.1a}$$

$$\frac{\partial^2\psi}{\partial X^2}+\frac{\partial^2\psi}{\partial Y^2}=-f\omega. \tag{2.1b}$$

Here, $\mathrm{Re}=2u_\infty l/\nu$ (l = half-chord of the ellipse), X and Y are the new space coordinates, f is the Jacobian of the mapping. Far away from the body, f can be approximated by 1. The origin is located at the center of the ellipse.

The system (2.1) is solved using a time-marching procedure: Given ψ and ω at time t, u and v are derived from (1.2). $\omega(t+\Delta t)$ is inferred from the advection–diffusion equation (2.1a). $\psi(t+\Delta t)$ is then computed by solving the Poisson equation (2.1b) with the value of ω at time $t+\Delta t$.

The method used here has been proposed and implemented by Ta Phuoc Loc

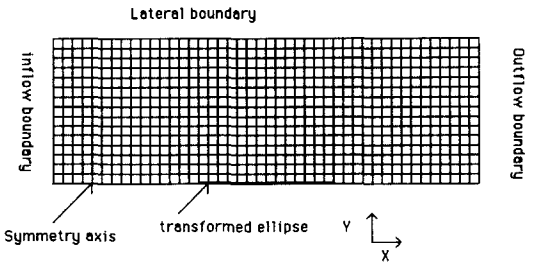


FIG. 3. Numerical grid.

and Daube [15]. We simply modified the inflow and outflow boundary conditions. It is a combination of two numerical schemes, a fourth order in the space compact method for the solution of Eq. (2.1b) and a second order one for Eq. (2.1a). We present now the outlines of the compact fourth order scheme. It is a compact Hermitian finite-difference scheme. The fourth order accuracy is achieved with a three-point approximation and by the introduction of the first and second derivatives of the unknown variables as the unknowns of the problem. This technique has been proposed by Collatz [16] and Kreiss (see Orszag and Israeli [17]) and developed by Hirsch [18] in the solution of some problems in fluid mechanics.

Let h denote the spatial step of discretization, and g_i, g_i', and g_i'', the values of the function g and its first and second derivatives at node i. The following tridiagonal relations can be written:

$$g'_{i-1} + 4g'_i + g'_{i+1} = 3(g_{i+1} - g_{i-1})/h + \mathcal{O}(h^4)$$
$$g''_{i-1} + 10g''_i + g''_{i+1} = 12(g_{i+1} - 2g_i + g_{i-1})/h^2 + \mathcal{O}(h^4).$$

Thus, one must impose boundary conditions not only for the unknown g, but also for its first and second derivatives and we have two types of boundary conditions. The first ones are related to the compact fourth order scheme and ensure the consistency of the numerical scheme. The second ones are related to the physical boundary conditions at infinity and we shall design open boundary conditions. The equations are solved by an ADI method; i.e., they are discretised by a finite difference method and the systems are solved by alternate sweeps over the columns and over the rows. The convergence is accelerated by the use of relaxation coefficients (see Wachpress [19]). For the computation of ω, an ADI algorithm is applied to solve the transport equation for the vorticity and an upwind scheme with second-order corrections is used for the discretization of the convective term.

4. Open Boundary Condition at Outflow

In [7], the authors designed open outflow boundary conditions for the fluid flow past a body in the velocity–pressure formulation and established error estimates and well-posedness theorems for the linearized equations. In a region sufficiently far from the body, the Navier–Stokes equations are linearized about the constant state at infinity. The resulting system can be solved explicitly in the Fourier space, which provides an integro-differential (in t and y) relation between u, v, and p at each point x. These relations are simpler than the linearized Navier–Stokes equations and are used as open boundary conditions. In a previous paper [6], the same work was done for the transport equation for vorticity and a series of open boundary conditions were designed. The first one is $\omega_t + u_\infty \omega_x = 0$, it is local in time and in space. The term $\omega_t + u_\infty \omega_x$ is the convection term of the linearized transport equation. Thus the nonlinear equivalent of the linear boundary condition is:

$$\omega_t + u\omega_x + v\omega_y = 0. \tag{3.1}$$

This nonlinear outflow boundary condition is the one we used and was already implemented in the code. Thus, we still have to design a new boundary condition for the stream function.

The adaptation of the previous works to the stream function vorticity formulation introduces two difficulties: the domain we consider here is bounded in the direction perpendicular to the flow and we have to design boundary conditions for the Poisson equation (2.1b). The former is easily (at least formally) overcome by considering eigenmodes expansion instead of the Fourier transform. The latter is much more serious, since the right-hand side in (2.1b) is not compactly supported in the computational domain and furthermore is not known outside Ω. Since we obviously do not want to compute $\omega(t+\Delta t)$ outside the computational domain (denoted Ω^T), we shall replace it by something easily expressed analytically. Taking advantage of the fact that we actually compute the steady state, we replace $\omega(t+\Delta t)$ outside Ω_t by taking the steady state ω^s. Let us make this more precise.

Since the computational domain is bounded in the y direction, it is necessary to have lateral boundary conditions. Let y lies between 0 and L. Consider a fluid flow around a body (domain Ω_c) in a slipping tube defined by $0 \leqslant y \leqslant L$ and $x \in \mathbb{R}$ (see Fig. 4). A slipping tube is a tube where fluid flows without dissipation at the wall of the tube. These properties yield the following boundary conditions at the wall of the tube:

At the solid wall, the tangential velocity and the tangential stress σ_{12} are zero. Since $\sigma_{12} = \rho(\partial u/\partial y + \partial v/\partial x)$ and v is zero, we have that $\partial u/\partial y = 0$. Thus, in the stream-function vorticity formulation, $v = 0$ is equivalent to

$$\partial\psi/\partial x = 0 \tag{3.2a}$$

and $\partial u/\partial y = 0$ and $\partial v/\partial x = 0$ imply

$$\omega = 0. \tag{3.2b}$$

The boundary conditions (3.2a) and (3.2b) at the wall of the tube being established, we are able to design an outflow boundary condition. The behavior at infinity is still given by (1.5).

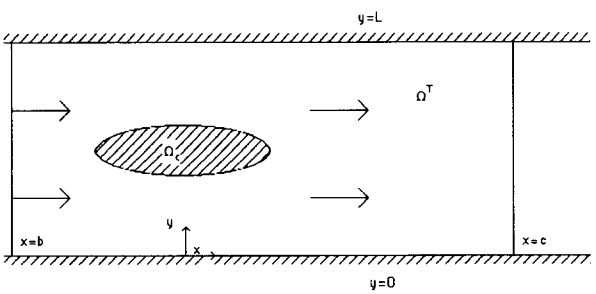


FIG. 4. Ellipse in the tube.

Let us consider a strip $b<x<c, 0<y<L$ and $\Omega^T=\{(x, y)/b<x<c$ and $0<y<L\}/\Omega_c$. If c is large enough, it is possible to linearize the equations outside Ω^T; f is approximated by 1 and the design of open boundary conditions can be studied by taking Eq. (1.4). Furthermore, in $\mathbb{R}^2/\Omega^T$ the flow is almost constant, equal to $u_\infty=a$ and $v_\infty=0$, a is assumed to be positive. In the stream function-vorticity formulation, we have: $\psi_\infty=ay+C^{te}$ and $\omega=0$.

Consider the linearized steady Navier–Stokes equations:

$$|a\omega_x=v\Delta\omega \tag{3.3a}$$

$$|\Delta\psi=-\omega, \qquad x\geqslant c \text{ and } 0\leqslant y\leqslant L \tag{3.3b}$$

with the following boundary conditions:

$$\omega(c, y)=\omega_0(y) \qquad \text{and} \qquad \psi(c, y)=\psi_0(y)$$

$$\omega=0 \text{ and } \psi=0 \qquad \text{at infinity}$$

$$\omega=0 \text{ and } \partial\psi/\partial x=0 \qquad \text{at} \quad y=0 \text{ and } y=L.$$

The compatibility of the boundary conditions is ensured by:

$$\omega_0=0 \qquad \text{at} \quad y=0 \text{ and } y=L$$

$$\psi_0=0 \qquad \text{at} \quad y=0 \text{ and } y=L.$$

Hence, the boundary condition at the wall of the tube is simply

$$\psi=0 \qquad \text{at} \quad y=0 \text{ and } y=L;$$

ψ and ω have the same boundary condition at the wall of the tube. Thus, (3.3) can be solved by separation of variables. Consider first the advection diffusion equation. ω is sought in the form $\sum\alpha_n(x)\,\varphi_n(y)$. Let $\alpha(x)\,\varphi(y)$ be a solution to (3.3). We have

$$\frac{-(a/v)\alpha'+\alpha''}{\alpha}=-\frac{\varphi''}{\varphi}=k,$$

where k is a real constant. Thus, φ has the form:

$$Ae^{i\sqrt{k}y}+Be^{-i\sqrt{k}y}.$$

The boundary condition at $y=0$ and $y=L$ gives $\varphi(0)=\varphi(L)=0$. The existence of nontrivial solutions is equivalent to:

$$k^{1/2}=n\pi/L \quad n\in\mathbb{N}^*.$$

Therefore, $\varphi=2iA\sin(n\pi y/L)$. Let

$$\varphi_n(y)=\sin(n\pi y/L). \tag{3.4}$$

It is well known that $(\varphi_n)_{n\geqslant 1}$ is an orthogonal base of $H_0^1(]0, L[)$. Therefore, ω can be sought in the form:

$$\omega(x, y) = \sum_{n\geqslant 1}^{\infty} \alpha_n(x)\, \varphi_n(y).$$

We shall need the expansion of ω_0 on the basis $(\varphi_n)_{n\geqslant 1}$,

$$\sum_{n\geqslant 1}^{\infty} \alpha_n^0 \varphi_n(y) = \omega_0(y);$$

ω is a solution to the advection diffusion equation and hence, α_n solves the second-order ordinary differential equation:

$$a\alpha_n'(x) - \nu\alpha_n''(x) - \nu\left(\frac{n\pi}{L}\right)^2 \alpha_n(x) = 0, \qquad x \geqslant c,\ n \geqslant 1$$

$$\alpha_n(c) = \alpha_n^0$$

$$\alpha_n(x) \to 0 \qquad \text{as} \quad x \to \infty.$$

It is easy to see that there is a unique solution $(\alpha_n)_{n\in\mathbb{N}^*}$ to this equation,

$$\alpha_n(x) = \alpha_n^0 e^{\lambda^-(n)(x-c)}$$

where

$$\lambda^-(n) = \left(a - \sqrt{a^2 + 4\nu^2\left(\frac{n\pi}{L}\right)^2}\right)\Big/ 2\nu$$

Thus,

$$\omega(x, y) = \sum_{n\geqslant 1}^{\infty} \alpha_n^0 e^{\lambda^-(n)(x-c)} \sin\left(\frac{n\pi}{L} y\right). \tag{3.5}$$

Once ω is given, ψ solves the Poisson equation:

$$|\Delta\psi = -\omega \qquad x \geqslant c,\ 0 \leqslant y \leqslant L$$

$$|\psi(c, y) = \psi_0(y)$$

$$|\psi(x, 0) = \psi(x, L) = 0$$

$$|\psi \to 0 \qquad \text{at infinity.}$$

At the wall of the tube, ψ is zero. Therefore, ψ can be expanded on the basis $(\varphi_n)_{n\geqslant 1}$:

$$\psi(x, y) = \sum_{n\geqslant 1}^{\infty} \gamma_n(x)\, \varphi_n(y). \tag{3.6}$$

We shall need the expansion of ψ_0 on $(\varphi_n)_{n\geqslant 1}$:

$$\sum_{n\geqslant 1}^{\infty} \gamma_n^0 \varphi_n(y) = \psi_0(y);$$

γ_n is a solution to the second-order differential equation

$$\gamma_n''(x) - \left(\frac{n\pi}{L}\right)^2 \gamma_n(x) = -\alpha_n(x) = -\alpha_n^0 e^{\lambda^-(n)(x-c)}, \qquad x \geqslant c,\ n \geqslant 1$$

$$\gamma_n(c) = \gamma_n^0$$

$$\gamma_n(x) \to 0 \qquad \text{as} \quad x \to \infty.$$

There is a unique solution to this equation:

$$\gamma_n(x) = \gamma_n^0 e^{-(n\pi/L)(x-c)} - \frac{\alpha_n^0}{\lambda^-(n)^2 - (n\pi/L)^2}\left(e^{\lambda^-(n)(x-c)} - e^{-(n\pi/L)(x-c)}\right) \tag{3.7}$$

which gives the value of ψ, given the values of ψ and ω at $x = c$.

Remark. The way we designed our boundary condition for ψ can be seen as a coupling between Navier–Stokes equations for $x \leqslant c$ and Oseen equations for $x \geqslant c$. Since the last equations can be solved explicitly, the coupling can be made on the outflow boundary at $x = c$.

The numerical implementation of the new boundary condition derived from (3.7) is the subject of the next section.

5. Numerical Implementation

We take advantage of the solving of Navier–Stokes equations by an ADI (see Section 2) method to use the following algorithm.

Algorithm. At each step of time, the Poisson equation for ψ is solved by an ADI methode, i.e., by alternate sweeps over the rows and over the columns of the grid. The outflow boundary condition can be implemented in a simple manner.

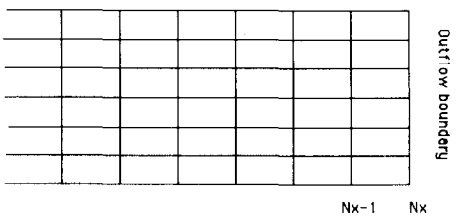


Fig. 5. Grid at outflow.

During a sweep over the columns:

(1) after the computation of the new values of ψ on the last but one column (column $Nx-1$, see Fig. 5), expand $\omega(t, Nx-1, \cdot)$ and $\psi(t, Nx-1, \cdot)$ on the base $\{\sin(n\pi y/L)\}_{n\geq 1}$.

(2) Calculate the coordinates of $\psi(t, Nx, \cdot)$ by formula (3.7).

(3) ψ on the last column Nx is calculated by summation of series (3.6).

Remark. In step (1), the coefficients are computed by identification on the column of the sum of the series with the values of the function at the nodes of the grid. Therefore, the number of coefficients is equal to the number of nodes on the grid. By using a rough Fourier transform, the open boundary condition increases the computational time by about 15%. As will be seen in the next section, this extra cost is compensated by the accuracy of the solution. Furthermore, the cost can be reduced by the use of fast Fourier transform.

6. Numerical Results

We computed the flow past an ellipse of excentricity 50% at $\mathrm{Re}=100$, using different types of inflow and outflow boundary conditions on ψ. Behind the ellipse, there is a large bubble wake followed by a wake (see Fig. 6).

For each type of computation, we use the boundary conditions for ω: At $x=b$: $\omega=0$ and at $x=c$: $\omega_t+u\omega_x+v\omega_y=0$.

For each type of computation, we use on the top boundary (wall of the tube), the boundary conditions of a slipping tube (i.e., $\omega=0$ and $v=0$), and on the symmetry axis, $\omega=0$ and $\psi=0$.

To test the open boundary condition, we made three types of computation using different types of inflow and outflow boundary conditions on ψ:

Type 1. A Dirichlet boundary condition, $\psi(b, y)=\psi(c, y)=\psi_\infty$, i.e., $u=u_\infty$.

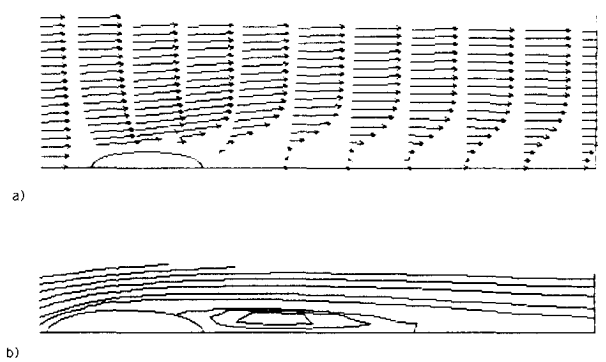


Fig. 6. Flow field near the ellipse: (a) velocity field; (b) streamlines.

Type 2. A Neumann boundary condition, $\partial\psi/\partial x = 0$ (i.e., $v = 0$) at $x = b$ and $x = c$.

Type 3. At $x = c$, we use the open boundary condition we have just presented. At $x = b$, since ω is taken to be zero it is easy to design an open boundary condition for the equation $\Delta\psi = 0$ by the same method we have used so far.

In any case, the boundary is located behind the bubble wake.

The results are compared with an "exact" solution. This solution is obtained by using outflow and inflow boundaries very far from the ellipse, on which are

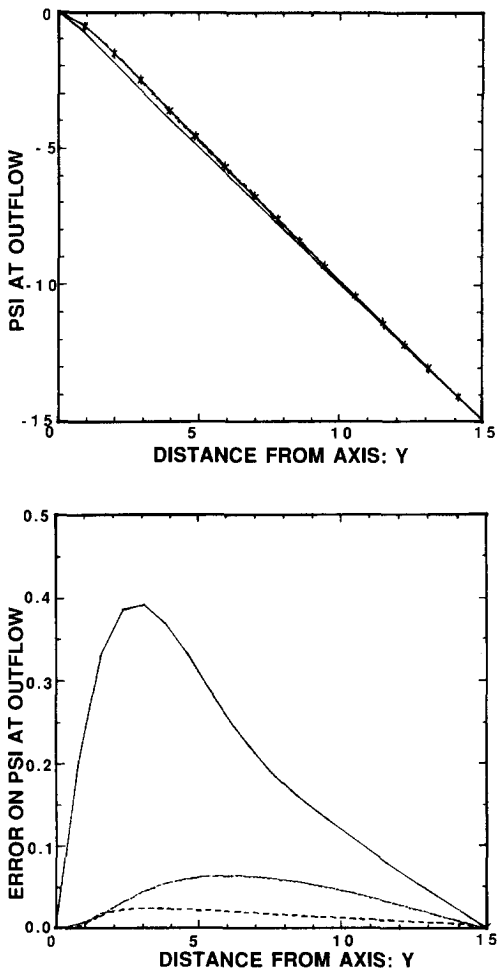


FIG. 7. Values and errors on ψ, u, v, and ω at outflow as a function of the distance to the symmetry axis for Dirichlet (———), Neumann (. . .), and transparent (– – –) boundary conditions (exact solution in ———).

prescribed open or Neumann boundary conditions (since the domain was quite large, the two solutions were very close to each other). The boundary cannot be located too close to the bubble wake, since otherwise the errors produced by any of the boundary conditions are too important. In our case, the boundary is located far enough from the bubble wake at $x = 10$. The ellipse is located between $x = -1.35$ and $x = 1.35$.

Comparison of the Different Boundary Conditions

As expected, the Dirichlet boundary condition proves to be very inaccurate (see Fig. 7 and Table I). The reason is that in the wake, the flow is very decelarated and that in the region outside the wake, the flow is still accelerated. Therefore, in the

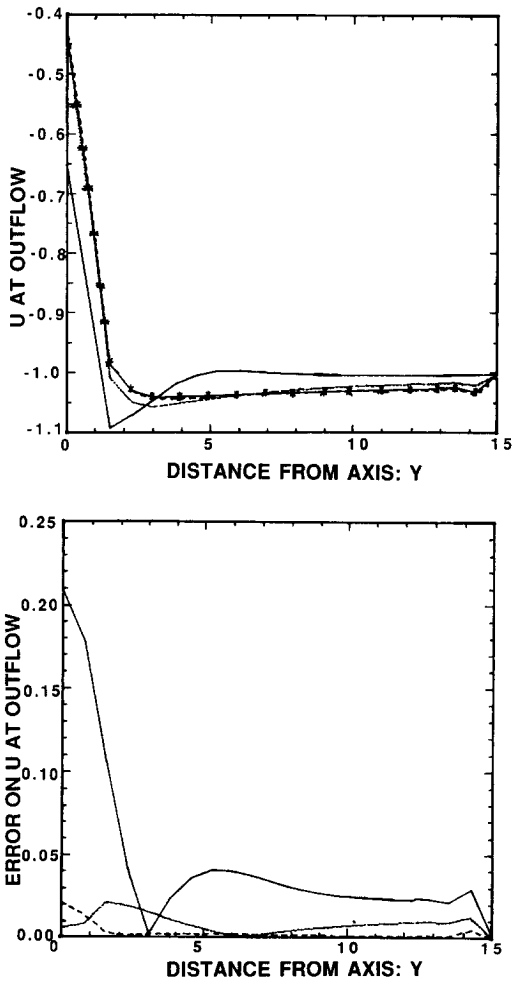
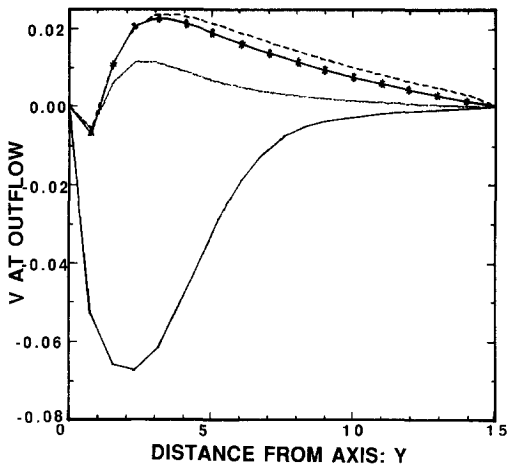


FIG. 7—*Continued*



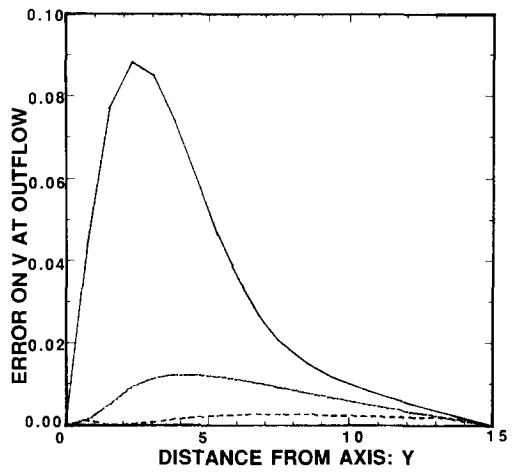


FIG. 7—*Continued*

TABLE I

Outflow Boundary at $x = 10$

Error in norm $L1(\%)$	Boundary conditions		
	Transparent	Neumann	Dirichlet
Stream function ψ	0,46	1.63	8.7
U	1.93	6.24	32.2

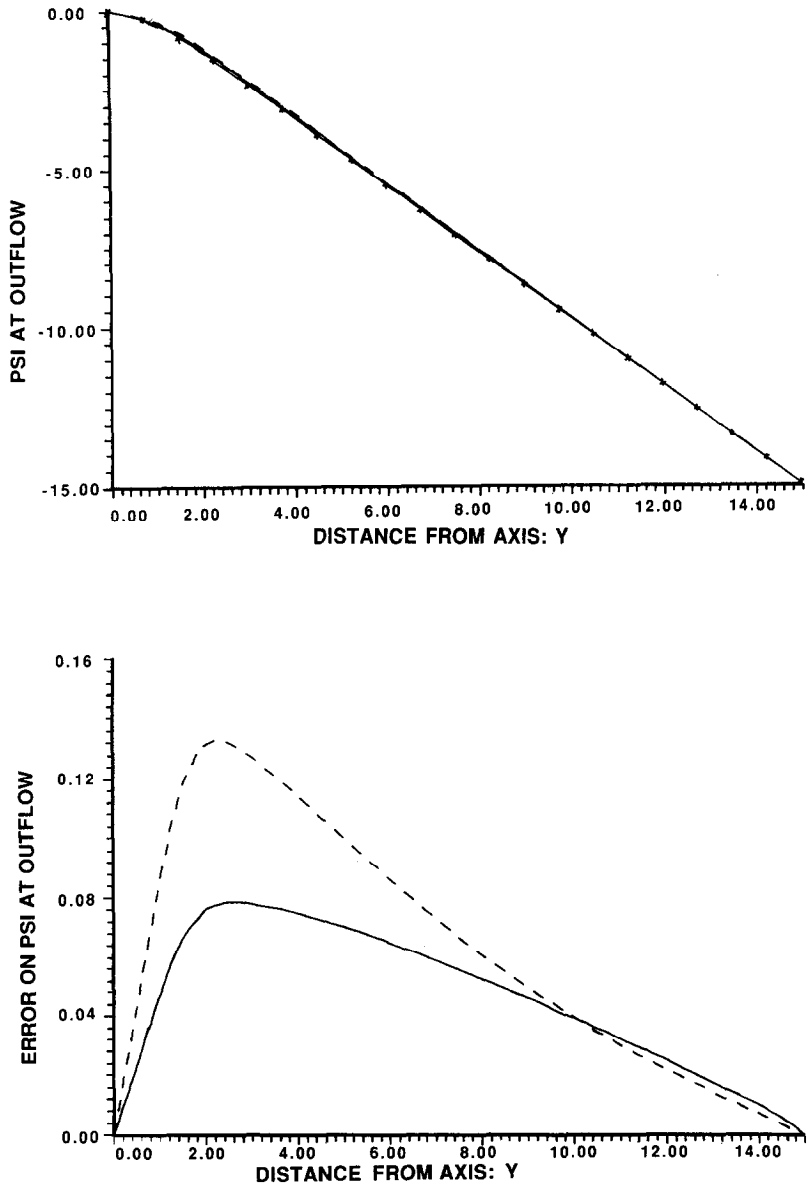
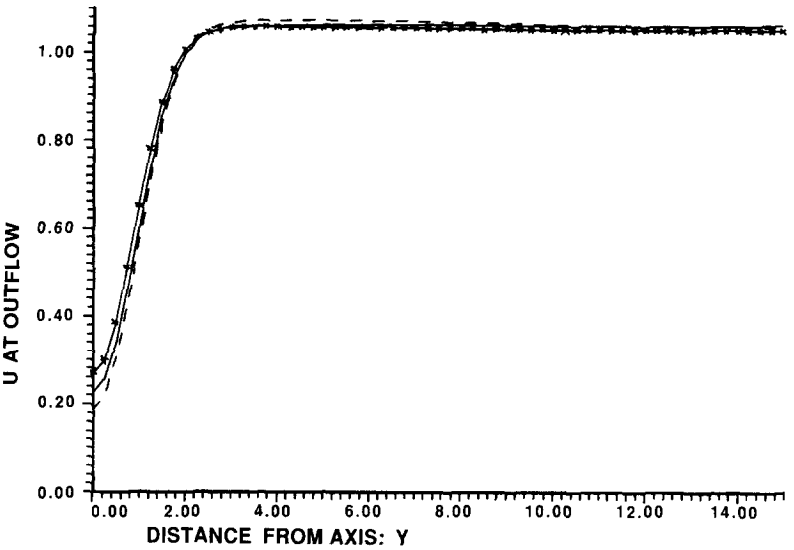


FIG. 8. Values and errors on ψ, u, v, and ω at outflow as a function of the distance to the symmetry axis for Neumann (---) and transparent (———) boundary conditions (exact solution is —·—·—).



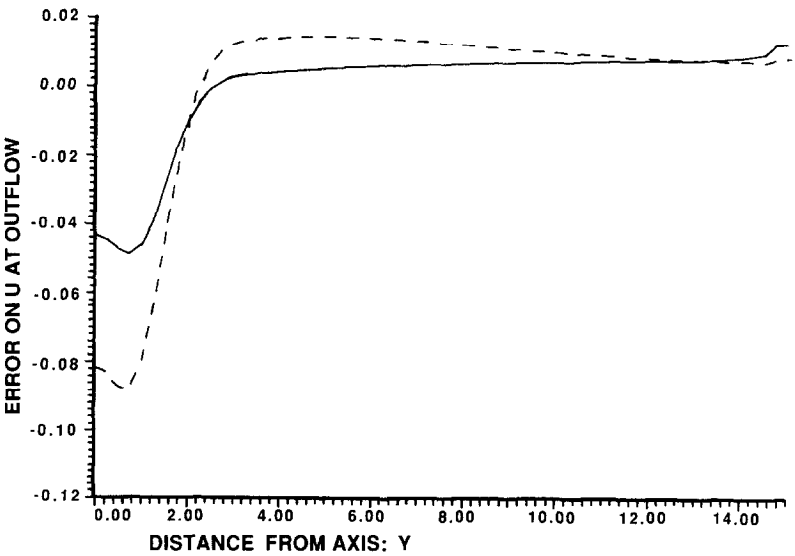
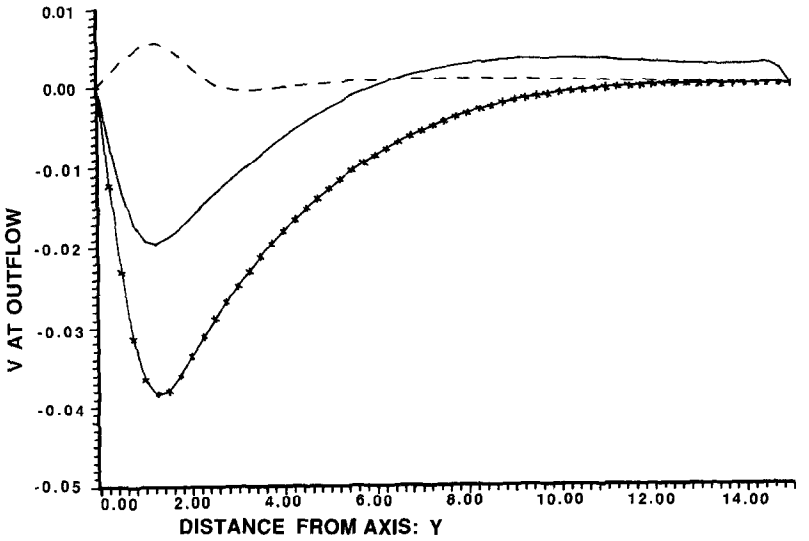


FIG. 8—*Continued*



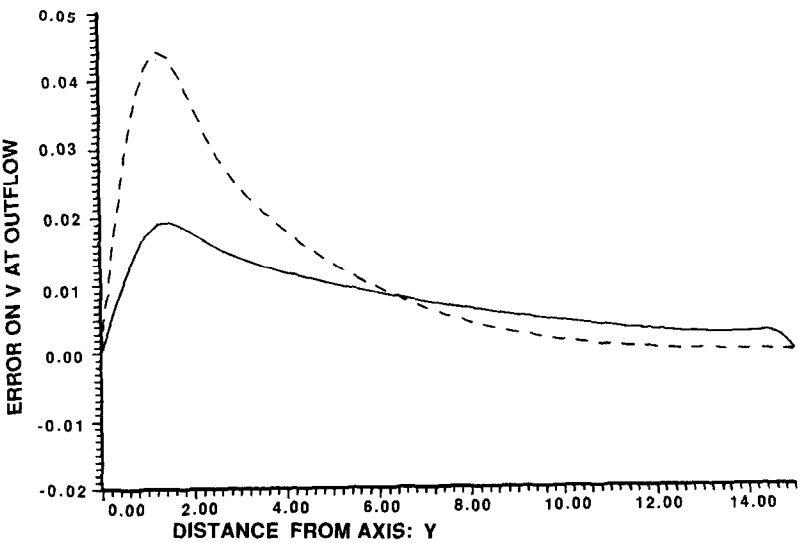
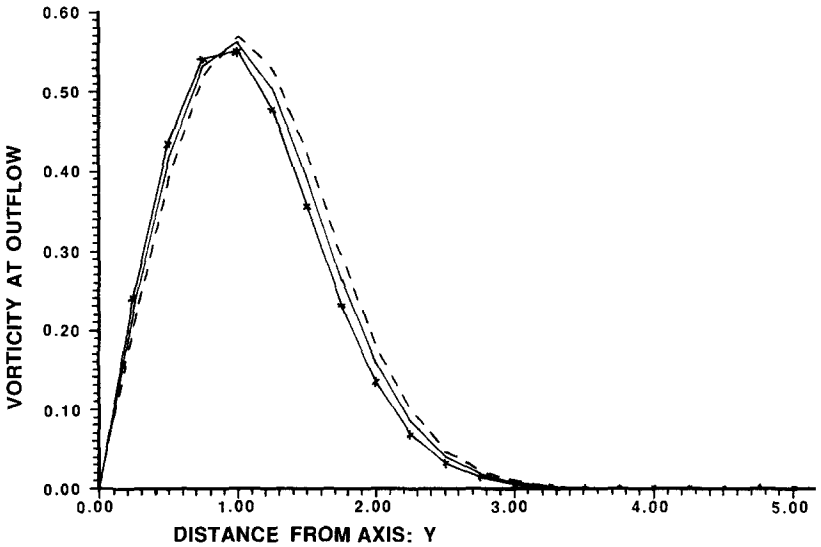


FIG. 8—*Continued*



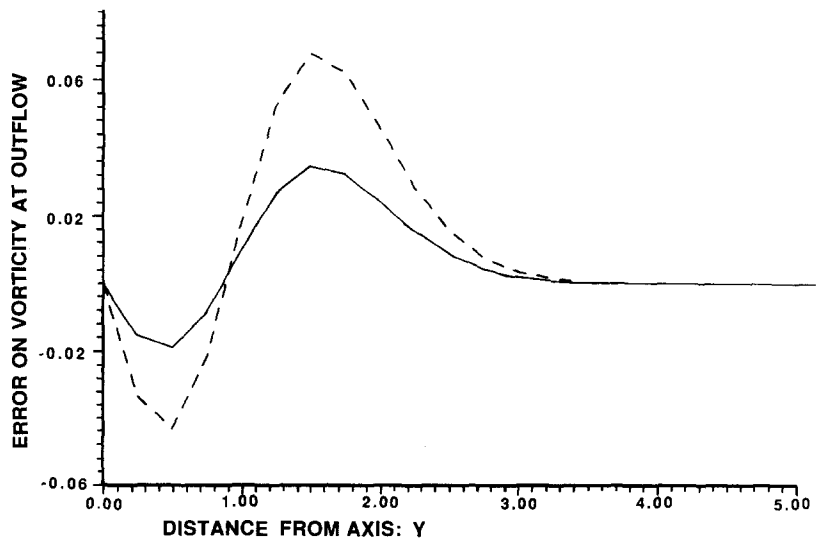


FIG. 8—*Continued*

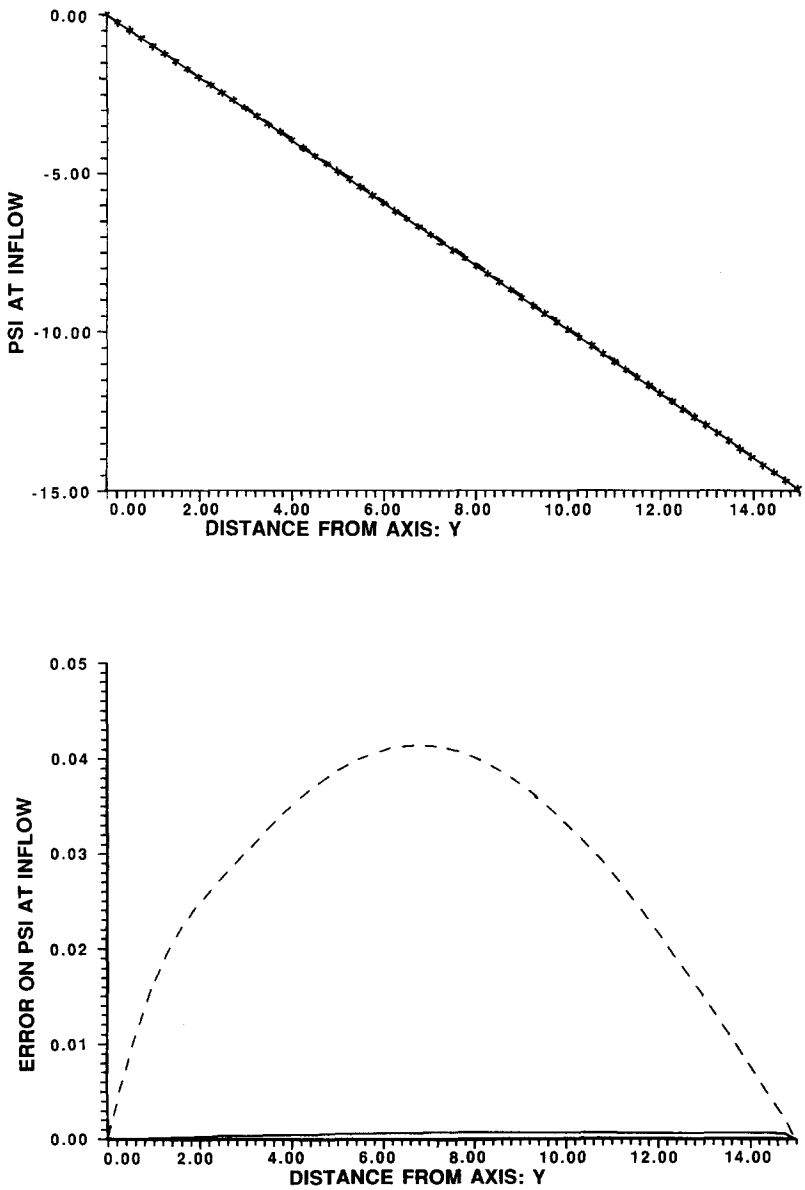
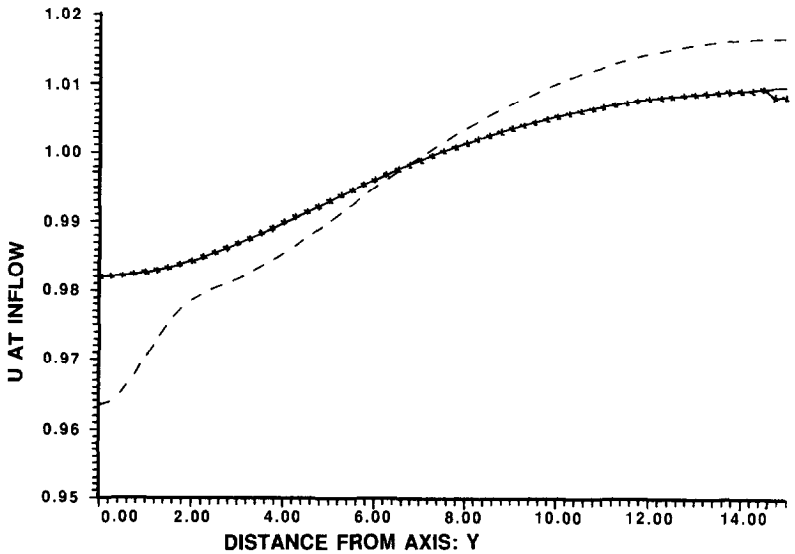


FIG. 9. Values and errors on ψ, u, v, and ω at inflow as a function of the distance to the symmetry axis for Neuman (---) and transparent (———) boundary conditions (exact solution is -·-·-·-).



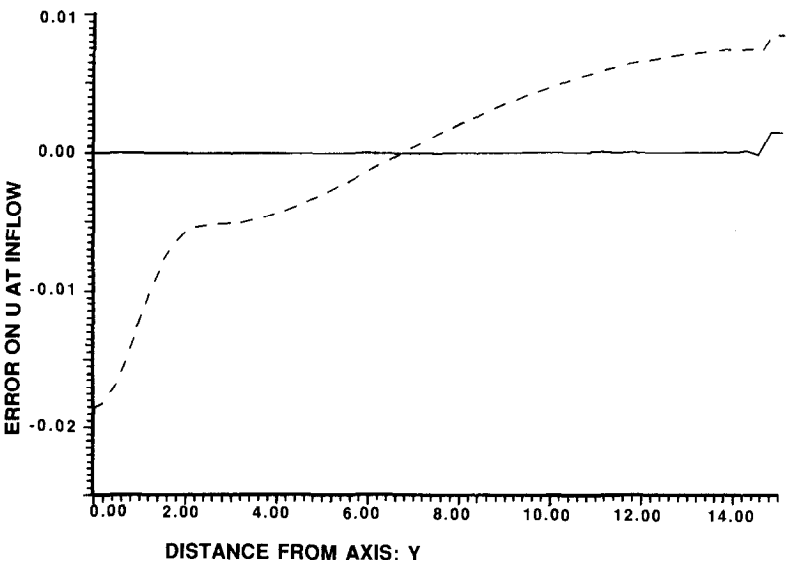
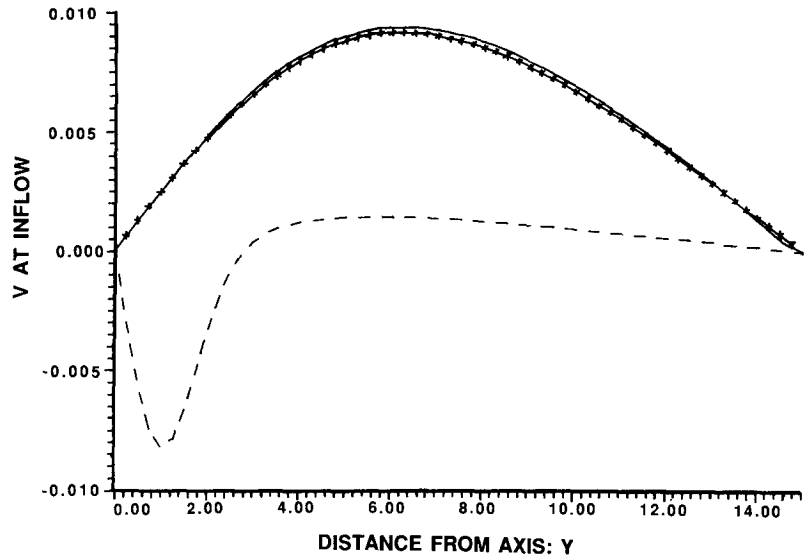


FIG. 9—*Continued*



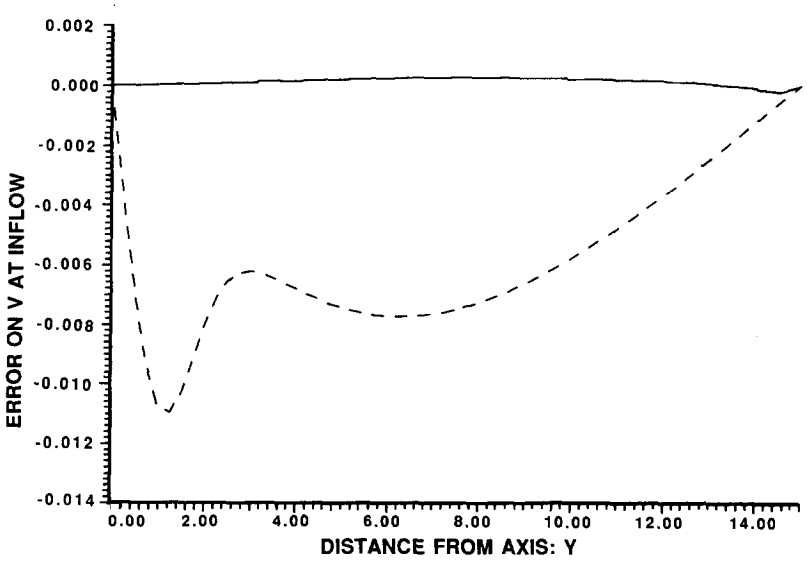


FIG. 9—*Continued*

following, we will only compare the last two boundary conditions. The open boundary condition proves to be more accurate than the Neumann boundary condition. The L^1 norm of the error is roughly three times as small. The error, near the outflow boundary (see Fig. 8), produced by the open boundary condition is twice as small as the one made with the Neumann boundary condition. The only exception is the error on v near the wall of the tube. The reason is most probably that in the Neumann case $v=0$ is prescribed at outflow and at the wall of the tube, while our open boundary condition is designed in a different way from the boundary condition at the wall of the tube, and their compatibility is not so well ensured in the implementation. At inflow $\omega=0$ and thus the open boundary condition becomes exact while the Neumann boundary condition still produces errors (see Figs. 9 and 10). Thus, it seems that an improvement of the open boundary condition should be to take into account the nonlinearity of the governing equations. This problem is quite difficult and does not seem to have been solved yet. In fact, it seems that it would be sufficient to linearize the equations around a state depending only on the direction perpendicular to the main flow, since in a wake the flow depends mainly on y only.

An other interesting point is the spatial dependance of the solution on the inflow and outflow boundary conditions. On Fig. 10, one sees that the error before the ellipse depends on the inflow boundary condition and that the error after the separation point depends mainly on the outflow boundary condition. But, from the front of the ellipse up to the separation ppoint, the error depends much less on the inflow and outflow boundary conditions. The boundary of the ellipse has a major influence. This can be related to the fact that in this region, the Prandtl equations can be applied. These equations depend only on an initially guessed pressure and using the pressure issued from a perfect fluid flow computation gives acceptable results.

7. Conclusion

A new boundary condition for the stream function is designed for the steady Navier–Stokes equations by using the linearized equations. This boundary condition is compared to the Dirichlet and Neumann boundary conditions. Our new boundary condition proves to be more accurate, especially for the Dirichlet boundary condition. And thus it is time saving, since for a given accuracy it is possible to use a smaller computational domain. It also has the advantage to have a firm mathematical basis (see [6, 7]). We also implemented a new boundary condition at inflow for the stream function. If one is interested in unsteady solutions, it would be necessary to design an open boundary condition for the unsteady Navier–Stokes equations.

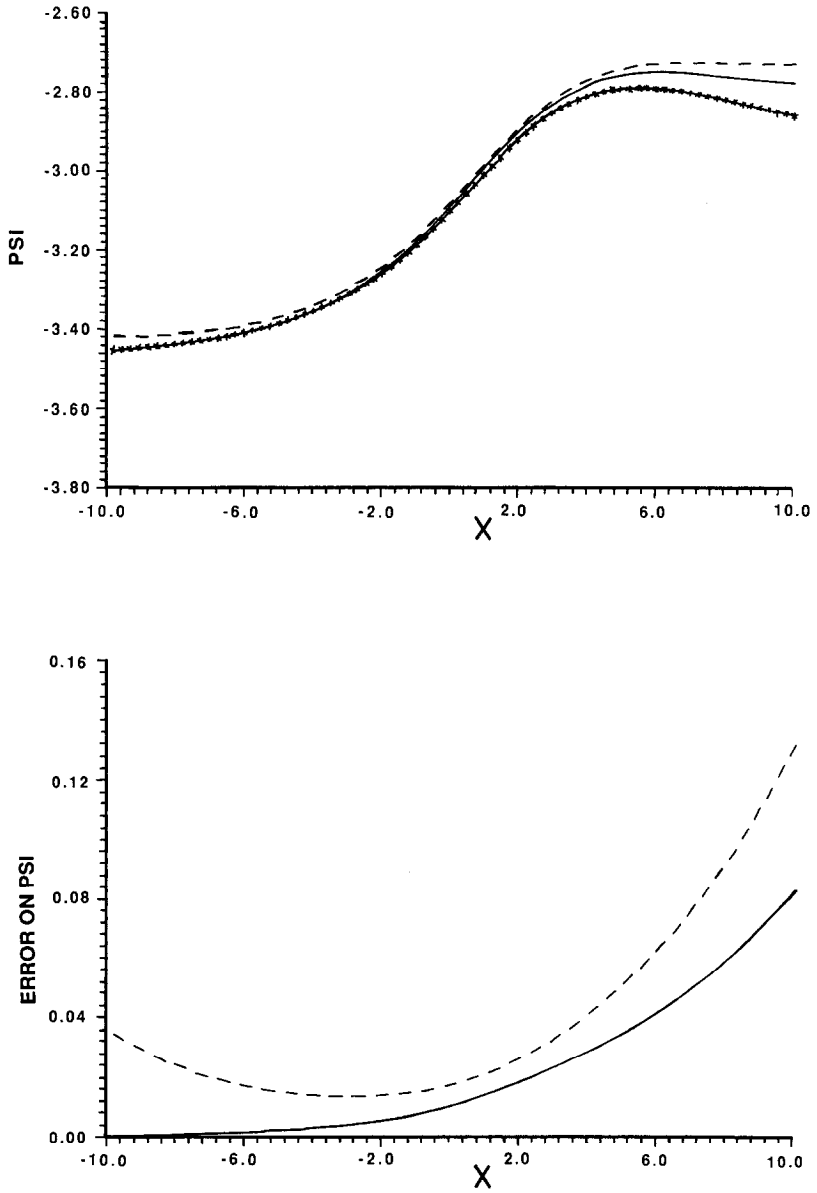


FIG. 10. Values and errors on ψ, u, v, and ω at $y = c^{st}$ for Neumann (----) and transparent (———) boundary conditions (exact solution is -·-·-·-).

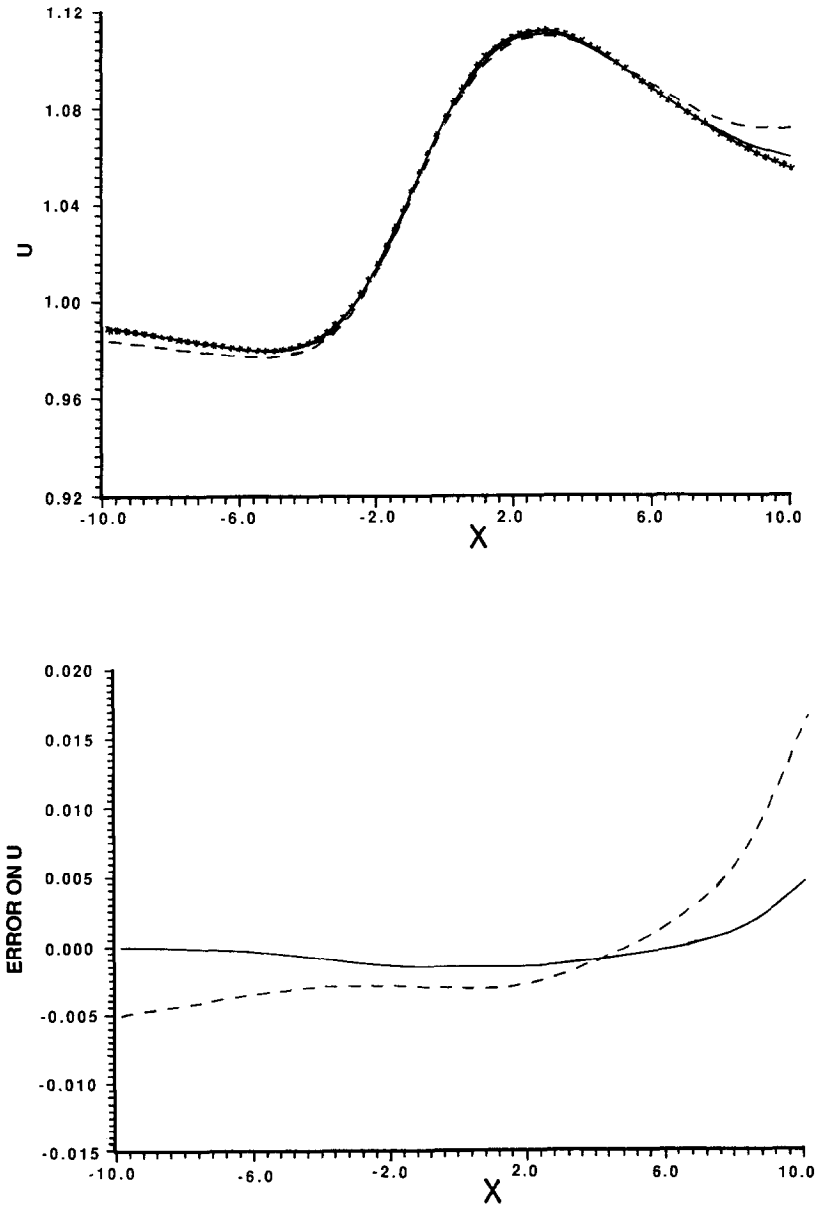


FIG. 10—*Continued*

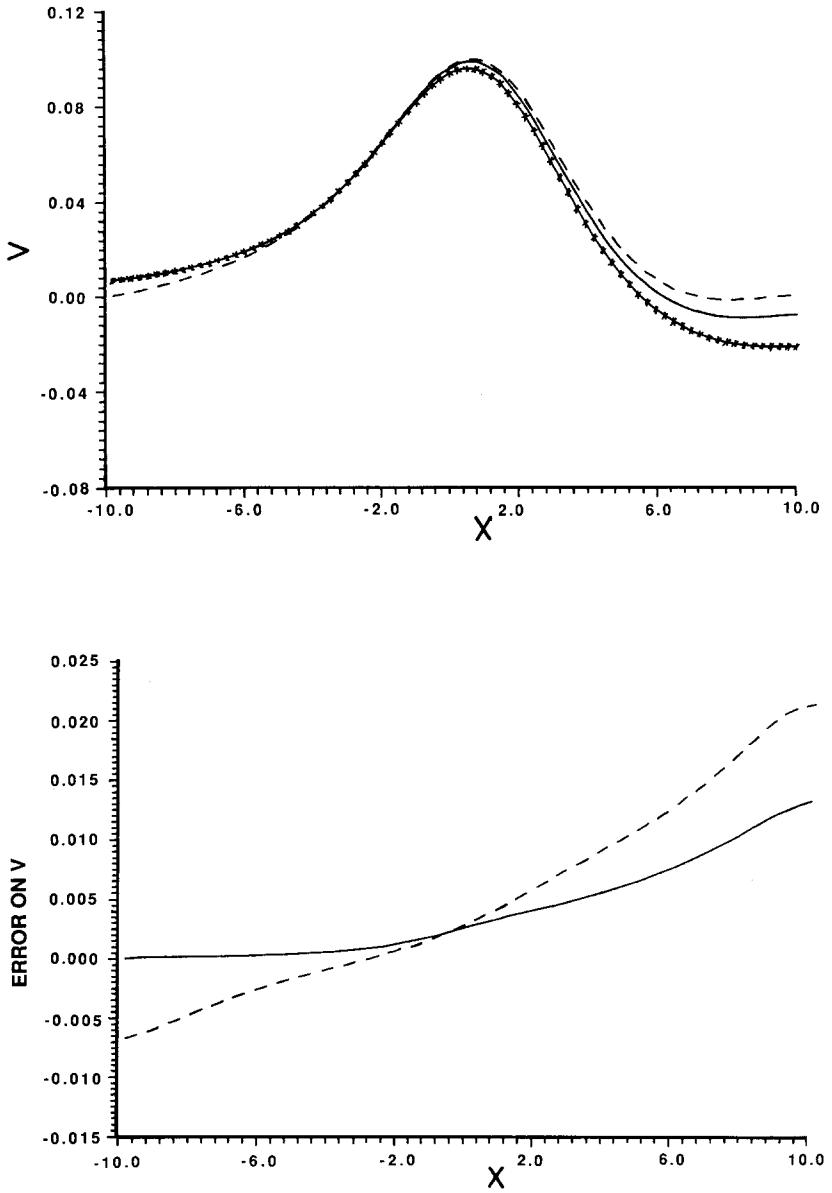


FIG. 10—*Continued*

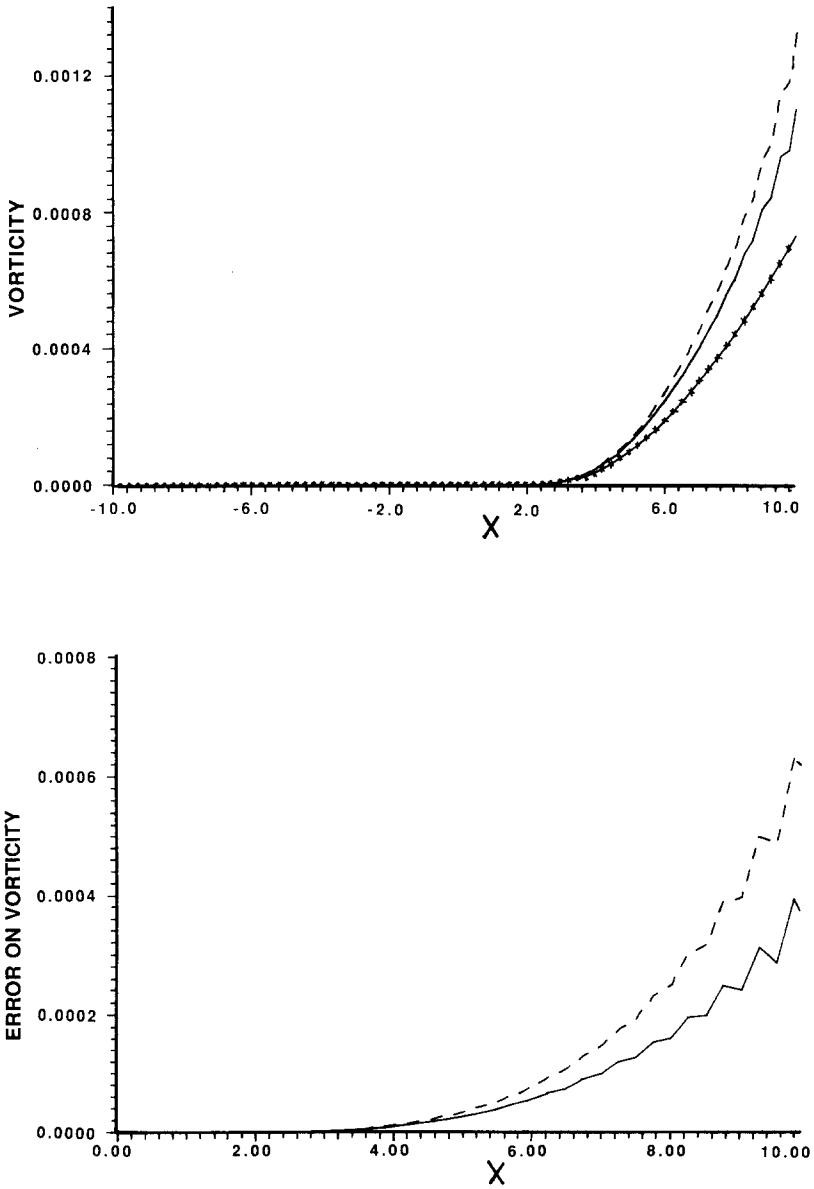


FIG. 10—*Continued*

ACKNOWLEDGMENTS

The author wishes to acknowledge the advices and encouragements of L. Halpern. Also he is very grateful to S. Huberson for his helpful discussions.

REFERENCES

1. G. MORETTI, *Phys. Fluids Suppl.* **11**, 13 (1969).
2. G. MORETTI AND M. ABBET, *AAIA J.* **4**, 2136 (1966).
3. Y. L. WANG AND P. A. LONGWELL, *AIChE J.* **10**, 323 (1964).
4. J. S. DEKRUIF AND A. A. HASSAN, *Comput. Fluids* **16**, 133 (1988).
5. W. S. VORUS, *J. Fluid Mech.* **132**, 163 (1983).
6. L. HALPERN, *Math. Comput.* **46**, 425 (1986).
7. L. HALPERN AND M. SCHATZMAN, *CR Acad. Paris*, SER. I **304**, 83 (1987).
8. C. JOHNSON AND J. C. NEDELEC, *Math. Comput.* **35**, 1063 (1980).
9. A. SEQUERIA, *Math. Methods Appl. Sci.* **5**, 356 (1983).
10. L. FERM, Report no. 102, Department of Computer Sciences, Uppsala University, 1985 (unpublished).
11. L. FERM AND B. GUSTAFSSON, *Comput. Fluids* **10**, 261 (1982).
12. M. LENOIR, *SIAM J. Num. Anal.*, inpress.
13. T. M. HAGSTRÖM, *J. Comput. Phys.* **69**, 69 (1987).
14. B. FORNBERG, *Boundary-Layer Separation*, edited by F. T. Smith and S. N. Brown (Springer-Verlag, New York/Berlin, 1987).
15. TA PHUOC LOC AND O. DAUBE, *Numerical Method for Laminar and Turbulent Flows* (Pineridge Press, Swansea, UK, 1985), p. 631.
16. L. COLLALTZ, *The Numerical Ttreatment of Differential Equations* (Springer-Verlag, New York/Berlin, 1986).
17. S. ORSZAG AND M. ISRAELI, *Annu. Rev. Fluid Mech.* **6** (1974).
18. R. S. HIRSCH, *J. Comput. Phys.* **19** (1975).
19. E. L. WACHSPRESS, *Iterative Solutions of Elliptic Systems* (Prentice–Hall, Englewood Cliffs, NJ, 1966).