JOURNAL OF COMPUTATIONAL PHYSICS 70, 373–396 (1987)

Numerical Solution for Viscous Flow for Two-Dimensional Domains Using Orthogonal Coordinate Systems

A. A. RANGWALLA AND B. R. MUNSON

Department of Engineering Science and Mechanics, Iowa State University, Ames, Iowa 50011

Received November 26, 1985; revised July 2, 1986

A method to generate two-dimensional orthogonal grids in simply and doubly connected domains is given. The method not only generates the grids but also finds the modulus of the domain simultaneously. Also, the Navier–Stokes equations are solved in some doubly connected domains. The insteady vorticity stream function approach is used. The stream function on one of the boundaries has to be updated at every time step. Comparisons are made between numerical and experimental results; quite good agreement can be achieved.

PART I

1. *Introduction*

Past research shows that the numerical solution of partial differential equations requires an accurate numerical representation of the boundary conditions. This can best be achieved by having a boundary conforming coodinate system. Such a method not only leads to simple application of boundary conditions (as far as programming is concerned) but also, in most cases, turns out to be very accurate. There are many approaches to generate boundary conforming coordinates; the most common among them is that developed by Thompson and others [1, 2]. However, one drawback of their method is that it does not guarantee orthogonal coordinates. Of course, for the numerical solution of partial differential equations, orthogonal coordinates are not essential, though in some cases highly desirable. For example, orthogonal coordinates can further simplify application of boundary conditions of the Neumann type. Another advantage is that they can lead to considerable simplifications in the finite difference algorithm. However, and most importantly, the accuracy of the solution can be improved [3].

In this article, we will describe a method to generate orthogonal coordinates in two dimensions and use these coordinates to solve the Navier–Stokes equations.

The classical method in two dimensions for the generation of orthogonal maps leads to conformal mappings. However, conformal maps are constrained to have equal scale factors in all directions, so that a small circle or square in one domain gets mapped onto a small circle or square in another domain, which may lead to

0021-9991/87 $3.00

unsuitable grids for the generation of numerical solutions [4]. To overcome this difficulty, some authors have used a constant scale factor in the whole domain [5, 6]. However, the scale factor cannot be known a priori but evolves as the solution is reached.

Mobley and Stewart [7], using monotonic functions to replace their conformal variables, pointed out a subtle but important concept of the conformal module. This concept has been elucidated in [8, 9] and will play a central role in our method, as will be seen in the next few sections. A further development occurs in [10], where the authors distinguish between two kinds of methods, namely the strong constraint method and the weak constraint method. Our method will be of the strong constrained kind where the shape of the domain is known. In [11] the weak constraint method of [10] was investigated. There it was found that the method works only for domains with a plane of symmetry. However, this is not too much of a disadvantage for many classes of domains.

In the present paper, we will develop a method to generate orthogonal coordinates in two dimensions for both simply and doubly connected domains. The boundary of the domains will be known but not the boundary correspondence. The grids for some of the doubly connected domains will then be used to solve the Navier–Stokes equations. This will be shown in Part II of this article along with some experimental results.

2. *Theory and Method*

This section deals with the governing equations for generating the grids and the numerical method for solving them. The mapping is restricted to simply or doubly connected domains in two dimensions. The equations for the simply connected domain will be developed first, followed by that for the doubly connected domain.

Consider a bounded, simply connected domain in (x, y) space (Fig. 1), which we want to map onto a rectangular region in (ξ, η) space. We want to achieve this so

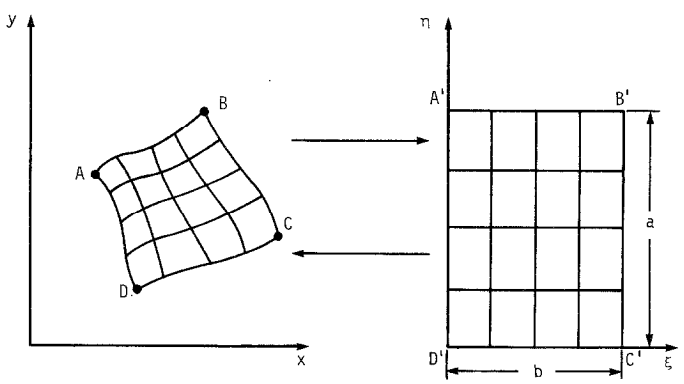

FIG. 1. Mapping of a simply connected domain in the (x, y) plane to a rectangle in the (ξ, η) plane with sides of length a and b.

that the points A, B, C, and D get mapped to the points A', B', C', and D', respectively. Also, we want the mapping to be orthogonal. Suppose $x(\xi, \eta)$ and $y(\xi, \eta)$ are mappings that satisfy the equations

$$\frac{\partial x}{\partial \xi} = K(x, y)\frac{\partial y}{\partial \eta} \tag{1a}$$

$$\frac{\partial x}{\partial \eta} = -\frac{1}{K(x, y)}\frac{\partial y}{\partial \xi}. \tag{1b}$$

These equations ensure that the mapping is orthogonal, where $K(x, y)$ is a function that determines the relative stretching in the map in the two directions. [$K(x, y)$ is restricted to be strictly greater than zero and bounded.] For the special case of $K(x, y) = 1$, Eqs. (1) become the Cauchy–Riemann equations. We do not directly solve Eqs. (1), but we solve a set of equations that can be derived from them and are second order. The equations we choose to solve are

$$\frac{\partial^2 x}{\partial \xi^2} + \frac{\partial^2 x}{\partial \eta^2} = \left(K - \frac{1}{K}\right)\frac{\partial^2 y}{\partial \xi\, \partial \eta} + \frac{\partial K}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{1}{K^2}\frac{\partial K}{\partial \eta}\frac{\partial y}{\partial \xi} \tag{2a}$$

$$\frac{\partial^2 y}{\partial \xi^2} + \frac{\partial^2 y}{\partial \eta^2} = \left(-K + \frac{1}{K}\right)\frac{\partial^2 x}{\partial \xi\, \partial \eta} - \frac{\partial K}{\partial \xi}\frac{\partial x}{\partial \eta} - \frac{1}{K^2}\frac{\partial K}{\partial \eta}\frac{\partial x}{\partial \xi}. \tag{2b}$$

Equation (2a) is obtained by differentiating Eq. (1a) with respect to ξ, Eq. (1b) with respect to η, and adding the results. Equation (2b) is derived similarly. Of course, there is no particular reason why the form of the equations chosen is that given by Eqs. (2). We could have chosen the formulation given in [10]. Equations (2) by themselves do not assure orthogonality. We have to apply the boundary conditions properly and take into account the modulus of the domain. The definition of the modulus follows. Since we want to "fix" four points, the ratio of the sides of the rectangle in the (ξ, η) plane cannot be arbitrary, but it must depend upon the domain and $K(x, y)$ (see [9]). This ratio can be considered the modulus of the domain and is given by

$$\frac{b}{a} = \iint \left(\frac{du}{d\eta}\right)^2 d\xi\, d\eta, \tag{3}$$

where $\nabla^2_{\xi,\eta} u = 0$, with $u = 0$ on $D'C'$, $u = 1$ on $A'B'$, and $\partial u/\partial \xi = 0$ on $A'D'$ and $B'C'$. In fact, $u = \eta/a$ and hence it is trivially true that Eq. (3) is an identity. Equation (3) by itself is not very useful, but it can be converted to

$$\frac{b}{a} = \iint \left(\frac{du}{dn}\right)^2 \frac{dA}{K(x, y)}, \tag{4}$$

where dA is the area of the image in the (x, y) plane of an elemental area in the (ξ, η) plane. The gradient of u, along lines of constant ξ, is given by

$$\frac{du}{dn} = \nabla_{x,y} u \times \frac{\nabla_{x,y} \eta}{|\nabla_{x,y} \eta|}.$$

The right-hand side of Eq. (4) would have to be evaluated numerically. Also, since $u = \eta(x, y)/a$, it is automatically known on lines of constant η. If we choose an $M \times N$ grid (i.e., i running from 1 to M and j running from 1 to N), then $\Delta\xi = b/(M-1)$ and $\Delta\eta = a/(N-1)$. Before we describe the steps in the numerical method, we define a few terms. By the phrase "interior of the domain," we mean those points (i, j), such that $2 \leqslant i \leqslant M-1$ and $2 \leqslant j \leqslant N-1$. By the phrase "interior of the interior," we mean those points (i, j), such that $3 \leqslant i \leqslant M-2$ and $3 \leqslant j \leqslant N-2$.

The numerical method can be described by the following steps:

(1) Choose an initial grid.

(2) Choose the ratio b/a.

(3) Iterate Eqs. (2a) and (2b) once in the interior of the domain. The iteration scheme is a simple iteration with relaxation. For points adjacent to the boundary, the boundary points in the iteration are replaced by Eqs. (1). That is, we have Neumann boundary conditions. The normal derivative is represented by one-sided, second-order, three-point differencing. The right-hand side is lagged by one step.

(4) After the third step, we have the positions of the grid points in the interior of the domain. Hence, we know the positions of the grid points just adjacent to the boundaries. Perpendiculars are then dropped from these adjacent grid points onto the boundary to obtain new boundary correspondence. For example, if one of the boundaries is $\xi = \text{constant} = \xi_0$, then the adjacent point would be $x(\xi_0 + \Delta\xi, \eta)$, $y(\xi_0 + \Delta\xi, \eta)$, and the perpendiculars from these points would be dropped onto the $\xi = \xi_0$ boundary. If more than one perpendicular could be dropped from a point, the shortest perpendicular should be chosen.

(5) With the new boundary correspondence, and with the help of Eqs. (1), obtain new points just adjacent to the boundary. Again, the normal derivative at the boundary is represented by one-sided, second-order, three-point finite differencing.

(6) Use the new points just adjacent to the boundary as Dirichlet conditions and iterate Eqs. (2a) and (2b) in the interior of the interior. (Maximum number of iterations used was 5. This number of iterations will be referred to as the number of "inner iterations.") Steps (5) and (6) seem to stabilize the method.

(7) Go back to step (3) if not converged or if the total number of iterations is less than some predefined number. This number will be referred to as the number of "outer iterations."

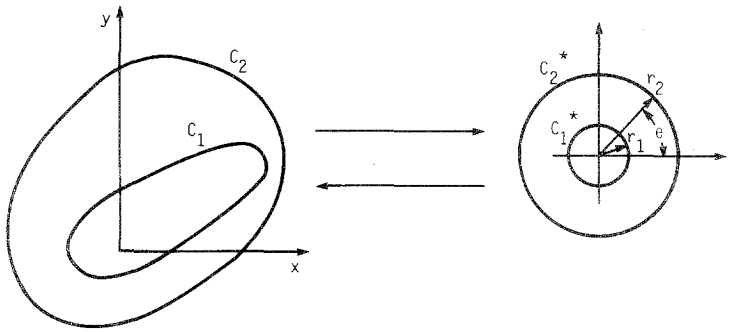

FIG. 2. Mapping of a doubly connected domain in the (x, y) plane to an annulus in the (r, θ) plane with inner radius r_1 and outer radius r_2.

(8) Check the ratio (or modulus) b/a by numerical integration of the right-hand-side of Eq. (4).

(9) If the new ratio does not match the old one, change it and go back to step (3); otherwise stop. Note that changing the ratio is the same as changing $\Delta\xi$ or $\Delta\eta$. (We can always keep one side fixed in length.) We have kept M and N constant.

For points just adjacent to the corner point, steps (4) and (5) are a bit different, as explained in the Appendix.

To form a grid in a doubly connected domain, the essential steps remain the same, but the details are a bit different. We map the region to an annulus as shown in Fig. 2.

The contour C_1 is mapped onto C_1^*, and the contour C_2 is mapped onto C_2^*. In this case, the equations for orthogonal maps are given by

$$\frac{\partial x}{\partial r} = \frac{K(x, y)}{r}\frac{\partial y}{\partial \theta} \tag{5a}$$

$$\frac{1}{r}\frac{\partial x}{\partial \theta} = -\frac{1}{K(x, y)}\frac{\partial y}{\partial r}. \tag{5b}$$

Equations (5), however, are not solved directly but are differentiated once more to obtain

$$\frac{\partial^2 x}{\partial r^2} + \frac{1}{r}\frac{\partial x}{\partial r} + \frac{1}{r^2}\frac{\partial^2 x}{\partial \theta^2} = \left(\frac{K}{r} - \frac{1}{Kr}\right)\frac{\partial^2 y}{\partial r\,\partial \theta} + \frac{1}{r}\frac{\partial K}{\partial r}\frac{\partial y}{\partial \theta} + \frac{1}{K^2 r}\frac{\partial K}{\partial \theta}\frac{\partial y}{\partial r} \tag{6a}$$

$$\frac{\partial^2 y}{\partial r^2} + \frac{1}{r}\frac{\partial y}{\partial r} + \frac{1}{r^2}\frac{\partial^2 y}{\partial \theta^2} = \left(\frac{1}{Kr} - \frac{K}{r}\right)\frac{\partial^2 x}{\partial r\,\partial \theta} - \frac{1}{r}\frac{\partial K}{\partial r}\frac{\partial x}{\partial \theta} - \frac{1}{K^2 r}\frac{\partial K}{\partial \theta}\frac{\partial x}{\partial r}. \tag{6b}$$

Again, as in the case of the simply connected domains, we have to properly apply the boundary conditions and take into account the modulus of the domain. We

cannot achieve the mapping, unless the ratio r_2/r_1 is the correct one. This ratio depends upon the domain that we are trying to map and the function $K(x, y)$. It is given by

$$\frac{r_2}{r_1} = e^{2\pi/D(u)}, \tag{7}$$

where

$$D(u) = \iint \left(\frac{du}{dr}\right)^2 r\, dr\, d\theta \tag{8}$$

and u satisfies

$$\nabla^2_{r,\theta} u = 0$$

with

$$\begin{aligned} u &= 0 \quad \text{on} \quad C_2^* \\ u &= 1 \quad \text{on} \quad C_1^*. \end{aligned} \tag{9}$$

Equation (9) implies that Eq. (7) is an identity. However, we can transform Eq. (8) to

$$D(u) = \iint K(x, y) \left(\frac{du}{dn}\right)^2 dA, \tag{10}$$

where dA is an elemental area in the (x, y) plane that is the image of the elemental area $r\, dr\, d\theta$, and du/dn is the gradient of u along lines of constant θ. The steps involved in the numerical application are the same as that for the simply connected case, except that now we have only two boundaries on which to drop perpendiculars. Also, the interior of the domain is all of the points that exclude the boundary points, whereas the interior of the interior is all of the points that exclude the boundary points and the points adjacent to the boundaries.

3. Numerical Equations and Results

We will briefly describe the finite difference formulation used. Since our scheme is a simple point iteration with relaxation, the formulation of the numerical equations is well known. The numerical analog of Eqs. (2a) and (2b) was an updated pointwise relaxation formula. The right-hand side of the equations was lagged by one step. The relaxation factor ω did play a role in the convergence of the scheme. We found that when the variation in $K(x, y)$ is large, we had to underrelax. The minimum underrelaxation factor used was 0.7. However, for cases where $K(x, y)$ is

nearly a constant, we could overrelax. An overrelaxation factor of 1.4 was used for all conformal cases and for all the doubly connected cases. However, we did not obtain an optimum ω by numerical experimentation. Also, the fact that for certain cases we had to underrelax does tell us that our numerical method might not be an optimum method, and that there is room for a large amount of improvement. But the point iteration method we have chosen does keep the programming very simple.

If the point (i, j) is adjacent to the boundary, we have to apply the Neumann boundary conditions. That is, if the point is $(2, j)$, then the numerical analog of Eq. (2a) is

$$\left(\frac{2}{3\Delta\xi^2}+\frac{2}{\Delta\eta^2}\right)x_{2,j}^{n+1}=\frac{1}{\Delta\eta^2}\left[x_{2,j+1}^{n}+x_{2,j-1}^{n}\right]+\frac{2}{3\Delta\xi^2}\left[x_{3,j}^{n}-K_{1,j}^{n}\left(\frac{y_{1,j+1}^{n}-y_{1,j-1}^{n}}{2\Delta\eta}\right)\Delta\xi\right]-RSX_{2,j}, \tag{11}$$

where $RSX_{2,j}$ is the numerical analog of the right-hand side of Eq. (2a) evaluated at $(2, j)$ at iteration level n. We have a similar equation for y. For a point next to the corner point, we need to apply the Neumann condition in both directions.

The numerical equations for doubly connected domains are similarly based upon Eqs. (6). Results will first be given for the doubly connected case, and then some results for the simply connected case will follow.

The first numerical experiment was tried on an annulus of inner radius 0.1 and outer radius 1. The function $K(x, y)$ was chosen to be 1. This is the conformal case where the mapping is known. The theoretical mapping is $10z$, where z is the complex variable, and the modulus $r_2/r_1 = 10$. The numerical results are given in Table I. One notable fact is that at every step the grid obtained is fairly orthogonal. However, this is because the geometry is admittedly quite simple and smooth. In some of the other examples that we tried, the grid at the intermediate steps was not always orthogonal. The numerical grid was quite accurate.

TABLE I

History of Convergence for the Case of an Annulus

Step	No. of Outer iterations	No. of inner iterations	Modulus r_2/r_1
1	9	3	1.01
2	6	3	4.5815
3	9	3	8.913757
4	4	3	10.39872
5	1	3	10.43659
9 (final step)	1	3	10.4356

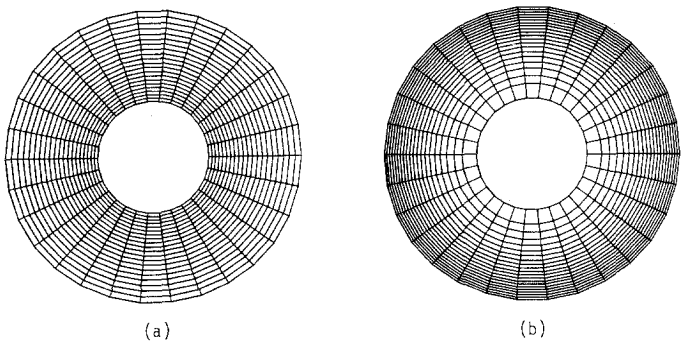

FIG. 3. Orthogonal grids in an annulus of inner radius 0.35 and outer radius 1: (a) $K(x, y) = 3$ and converged modulus is 1.399945, (b) $K(x, y) = \frac{1}{3}$ and converged modulus is 20.35164.

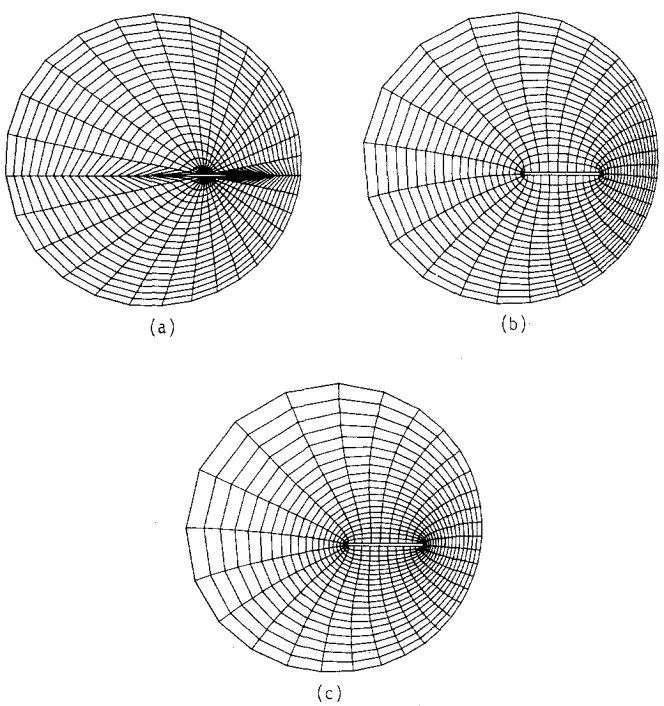

FIG. 4. Grid in a doubly connected domain—outer boundary is circular and inner body is a tab with rounded ends: (a) initial grid, (b) converged grid for $K(x, y) = 1$; converged modulus is 6.330519, (c) converged grid for $K(x, y) = 3$; converged modulus is 1.851812.

The discrepancy between the numerical modulus and the exact is due to the discretization involved. An error analysis of the first oder of Eq. (7) gives

$$\Delta\left(\frac{r_2}{r_1}\right)=\frac{-2\pi}{\{D(u)\}^2}\Delta(D(u))\frac{r_2}{r_1}. \tag{12}$$

For this case, $e^{2\pi/D(u)}$ is approximately equal to 10. Hence, $D(u)\cong 2.728$. Assuming a 1% error in evaluating $D(u)$, we find that $\Delta(r_2/r_1)\cong 0.23$, which is the same order of magnitude as the error we obtained in evaluating the modulus.

Numerical results of other doubly connected cases are shown in Figs 3a, b and Figs. 4a, b, c.

In Fig. 3, we have evaluated orthogonal grids for an annular region, where $K(x, y)$ is not equal to one. We find that when $K(x, y)>1$, the grid points are moved inward, and when $K(x, y)<1$, the points are moved outward. This does give us one method for clustering the points at a boundary.

Figure 4 is an example of a domain in which the Navier–Stokes equations were solved—a circular outer boundary with a thin tab inner boundary. This domain is not as trivial as the annulus. Since the inner body is so slender, the mapping does show large gradients at certain points. However, this did not seem to cause any problems. Figure 4a shows the initial grid for both cases. This grid is quite skewed, and the circumferential points on the inner body were purposely "clustered" at the wrong locations. Figure 4b shows the converged grid for $K(x, y)=1$, and Fig. 4c shows the converged grid for $K(x, y)=3$. As is apparent, choosing $K(x, y)$ greater than 1 does cluster the points toward the inner body, a very helpful feature for the kinds of flow we will be looking at in the second part.

Numerical results of simply connected domains are shown in Figs. 5a, b, and Figs. 6a, b, and c. Once again, we find that for those cases where the analytical solution is known, the numerical solution agrees quite well. In Fig. 5a, we have constructed a grid in a rectangular domain of sides 1.5 and 1, respectively. Now, for $K(x, y)$ equal to 1, the modulus of this domain is 1.5, and the grid is a uniform grid. Our numerical scheme gave us a modulus of 1.5009 and a grid that deviated very little from the theoretical. We then changed $K(x, y)$ to 3 for the same number of grid points. As expected, the modulus changed by a factor of 3, but the grid looked the same. This was to be expected. In fact, in [7] this fact is used when mapping all simply connected domains to an $M\times N$ rectangle. That is, if we have some $K(x, y)$ (need not be a constant) and we want to map the domain to any arbitrary $M\times N$ rectangle, all we have to do is find the right value of S such that the mapping generated by $SK(x, y)$ will satisfy the constraints of fixing four corner points. The value of S is not known, but it can be obtained as a part of the iteration scheme as pointed out in [7]. We could have done the same, but we wanted to use the same approach as the doubly connected case.

In Fig. 5b, we have constructed a grid for an arbitrary quadrilateral. The main features of this domain are the nonorthogonal corner points. These had to be specially handled (see Appendix). One thing we do note is the loss in orthogonality

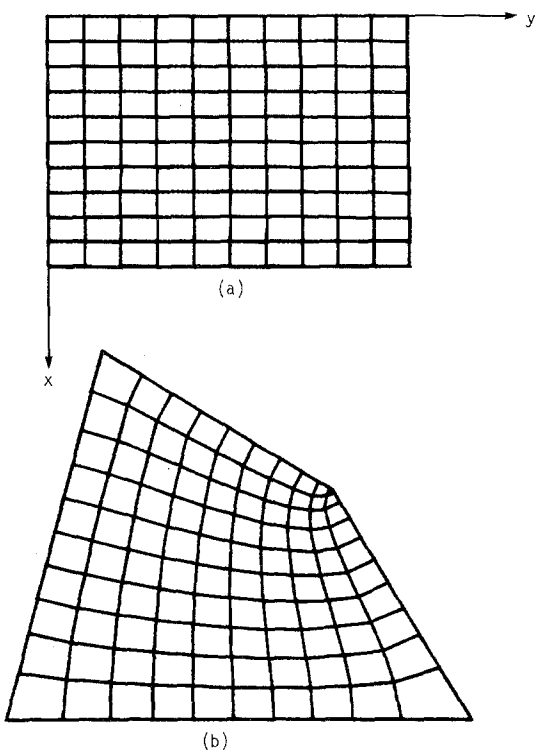

FIG. 5. Grids in some simply connected domains: (a) rectangular domain, $K(x, y) = 1$; converged modulus is 1.5009, (b) simply connected domain, $K(x, y) = 1$; converged modulus is 0.872.

near these corner points. This is expected because near the corner points, the constant ξ and η lines have high curvature, and the coarse discretization cannot approximate it accurately enough. Figures 6a, b, and c are examples of grids for different $K(x, y)$ values.

In Fig. 5b the maximum deviation from orthogonality is about 30°. This happens at the grid point closest to the obtuse corner. Near this corner the constant ξ and constant η lines have very high curvatures. The angles, however, are formed by straight line segments joining the grid points. In Figs. 6a, b, and c the maximum deviations from orthogonality are 18°, 20°, and 34°, respectively. The largest value occurs for two reasons. First, the right-hand corner point is nonorthogonal. Second, the relative stretching in the grid is such that the points are moved away from the top boundary. The constant ξ lines have a curvature reversal near the right-hand corner. This reversal has not been adequetely resolved.

Also, numerous other examples were tried for both the simply connected domain and the doubly connected ones. In all cases, quite good grids resulted. A final explanation is due. The grids were plotted on a Tektronix Terminal, and the hard

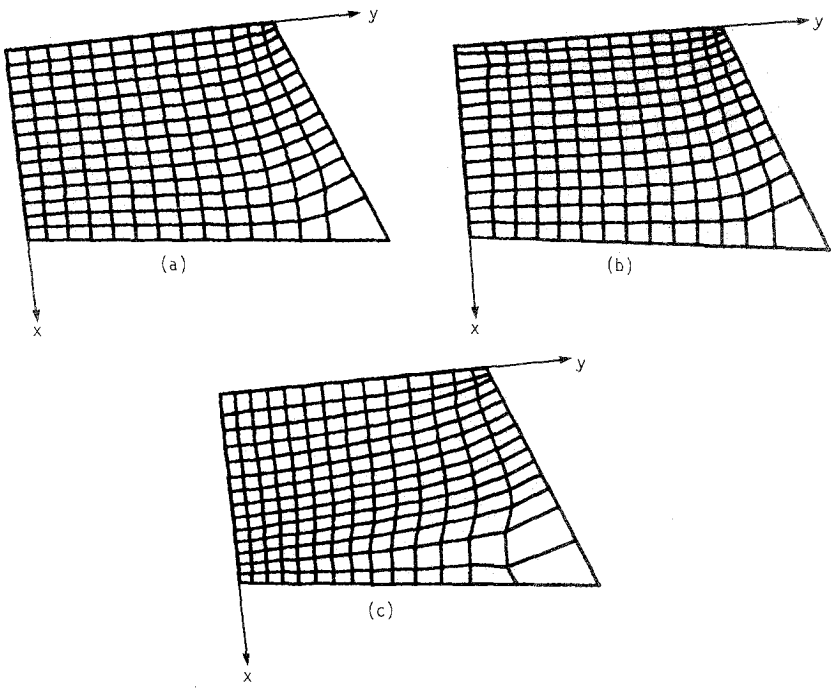

FIG. 6. Grids in a four-sided region for different $K(x, y)$: (a) $K(x, y) = 1$; converged modulus is 1.550257, (b) $K(x, y) = x + y + 1$; converged modulus is 3.52179, (c) $K(x, y) = 1/(x + y + 1)$; converged modulus os 0.6981274.

copies were made straight from the terminal. The scales on the x and y axes were not the same. That is the reason our figures seem to be shorted a bit in the vertical axis.

4. CONCLUSION

In this first part, we have presented a numerical technique to generate orthogonal maps. The problems we have essentially solved are the strong constraint as defined in [10], where the boundary of the domain is known. Unlike some of the previous methods, we have not used any Dirichlet boundary conditions. Also, the method seems to work quite well for highly skewed initial grids.

PART II

1. *Introduction*

In this part, we consider the numerical solution of the Navier–Stokes equations (NS equations) in a two-dimensional doubly connected domain. Also, comparisons with some experiments will be made.

We employ the vorticity-stream function formulation of the NS equations. This approach has several advantages (see [12]): the main one being the elimination of the pressure term. However, this formulation is useful only for those domains where the complete boundary correspondence for the stream function is known. For domains where the value of the stream function is not known a priori on one of the boundaries, but has to be obtained from the solution, this method at first glance does not seem appropriate. Consider the flow in an annulus with the outer wall rotating and the inner one fixed. We can fix the value of ψ on one boundary but cannot do so on the other. The value of ψ on the other boundary depends upon the velocity of rotation of the outer boundary and the geometry. In this paper, an attempt has been made to solve problems of this nature.

In [13] Thomas and Szewczyk encountered a similar problem. They solved a two-dimensional flow over a cylinder and did not fix the value of the stream function on the top boundary but allowed it to change. They achieved this by recalculating the inlet velocity at every step and then finding the total mass flow through the domain. For this they had to assume that $\partial\psi/\partial x$ was zero at the inlet. However, in the doubly connected domains that we consider, we cannot do this. The correct boundary condition for ψ is connected to the constraint that $\oint_c dp = 0$, where c is any closed contour and p is the pressure.

Admittedly, there have been numerous schemes utilizing the velocity stream function approach [14–17], where the implementations of the boundary conditions have varied from author to author. The results are also quite different. We have not made any formal attempt at evaluating the varying methods to see which works best. The choices we made were governed by simplicity and convenience in programming.

2. *Governing Equations*

The Navier–Stokes equations can be written as

$$\nabla^2\psi = \zeta \tag{13a}$$

$$\nabla^2\zeta = \mathrm{Re}\left[u\frac{\partial\xi}{\partial x} + v\frac{\partial\zeta}{\partial y}\right] + \mathrm{Re}\,\frac{\partial\zeta}{\partial t}, \tag{13b}$$

where Eqs. (13) are written in cartesian coordinates, ψ is the stream function, Re is the Reynolds number, and ζ is the vorticity. These equations can be transformed into the computational coordinates (ξ, η) to give

$$\alpha\psi_{\xi\xi} + 2\beta\psi_{\xi\eta} + \gamma\psi_{\eta\eta} + \delta\psi_{\xi} + \varepsilon\psi_{\eta} = \zeta \tag{14a}$$

$$\begin{aligned}\alpha\zeta_{\xi\xi} + 2\beta\zeta_{\xi\eta} + \gamma\zeta_{\eta\eta} + \delta\zeta_{\xi} + \varepsilon\zeta_{\eta} \\ = \mathrm{Re}[u(\xi_x\zeta_{\xi} + \eta_x\zeta_{\eta}) + v(\xi_y\zeta_{\xi} + \eta_y\zeta_{\eta})] + \mathrm{Re}\,\zeta_t,\end{aligned} \tag{14b}$$

where subscripts denote differentiation. In Eqs. (14), the coefficients are given by

$$\begin{aligned}
\alpha &= \xi_x^2 + \xi_y^2 \\
\beta &= \xi_x \eta_x + \xi_y \eta_y \\
\gamma &= \eta_x^2 + \eta_y^2 \\
\delta &= \xi_{xx} + \xi_{yy} \\
\varepsilon &= \eta_{xx} + \eta_{yy} \\
u &= -\frac{\partial \psi}{\partial y} = -[\xi_y \psi_x + \eta_y \psi_\eta] \\
v &= \frac{\partial \psi}{\partial x} = [\xi_x \psi_\xi + \eta_x \psi_\eta].
\end{aligned} \tag{15}$$

In evaluating the above, use is made of

$$\xi_x = \frac{y_\eta}{J}, \qquad \xi_y = \frac{-x_\eta}{J}, \qquad \eta_x = \frac{-y_\xi}{J}, \qquad \eta_y = \frac{x_\xi}{J},$$

where $J = x_\xi y_\eta - x_\eta y_\xi$ is the Jacobian of the transformation. All the differences in the above are central differences. The grid was generated by the method described in Part I, and the computational domain (ξ, η) for the solution of the NS equations was the index of the grid points. That is, point (x_{ij}, y_{ij}) of the grid was mapped onto $\xi = i$ and $\eta = j$. The coordinate ξ runs in the circumferential direction and the coordinate η runs from the inner to the outer boundary. The tangential velocity on the outer circular wall was normalized to one. Note that in Eq. (14) we have not used the fact that the grid is orthogonal.

The boundary conditions on the stream function ψ were derived by the condition of no slip and the fact that the boundaries are impermeable. Also, the grids are orthogonal to the boundaries. Hence, we have

$$\psi = 0 \qquad \text{at} \quad \eta = 1 \tag{16a}$$

$$\psi = \text{constant} \quad \text{(undetermined)} \qquad \text{at} \quad \eta = n \tag{16b}$$

$$\frac{\partial \psi}{\partial n_1} = 0 \qquad \text{at} \quad \eta = 1 \tag{16c}$$

$$\frac{\partial \psi}{\partial n_1} = 1 \qquad \text{at} \quad \eta = n, \tag{16d}$$

where $\partial / \partial n_1$ is the gradient in the direction normal to the boundary. Also,

$\psi_{1,\eta} = \psi_{m,\eta}$, since points $(1, \eta)$ and (m, η) are the same points in the physical plane. The boundary condition on vorticity has to be updated at every time step by

$$\zeta_{\text{wall}} = \frac{\partial^2 \psi}{\partial n_1^2} + \frac{1}{\rho} \frac{\partial \psi}{\partial n_1}, \tag{17}$$

where ρ is the radius of curvature of the boundary.

Before we elaborate upon the numerical equations, we will make a short note on the governing equations. For low Reynolds numbers, that is $\text{Re} \to 0$, Eqs. (13) reduce to the steady Stokes equations. Since it has been the experience of many that the unsteady equations are easier to evaluate numerically to convergence, for low Reynolds number (Reynolds numbers <1), we instead scaled them by R^2/ν (R is outer radius) to obtain

$$\nabla^2 \zeta = \text{Re}\left[u \frac{\partial \zeta}{\partial x} + v \frac{\partial \zeta}{\partial y} \right] + \frac{\partial \zeta}{\partial \tau} \tag{18a}$$

$$\nabla^2 \psi = \zeta. \tag{18b}$$

Hence, as $\text{Re} \to 0$, we obtain

$$\nabla^2 \zeta = \frac{\partial \zeta}{\partial \tau}, \qquad \nabla^2 \psi = \zeta,$$

which is the set of equations solved for Stokes flow.

3. *Numerical Equations*

The numerical scheme involved the solution of the following numerical equations. The numerical analog of Eq. (18a) in the (ξ, η) system was

$$\begin{aligned}
&A_{ij}(\zeta^n_{i+1,j} - 2\zeta^{n+1}_{i,j} + \zeta^{n+1}_{i-1,j}) \\
&\quad + \frac{B_{ij}}{4} \zeta^n_{i+1,j+1} - (\zeta^n_{i+1,j-1} - \zeta^{n+1}_{i-1,j+1} + \zeta^{n+1}_{i-1,j-1}) \\
&\quad + C_{ij}(\zeta^{n+1}_{i,j+1} - 2\zeta^{n+1}_{i,j} + \zeta^{n+1}_{i,j-1}) + \frac{D_{ij}}{2}(\zeta^n_{i+1,j} - \zeta^{n+1}_{i-1,j}) \\
&\quad + \frac{E_{ij}}{2}(\zeta^{n+1}_{i,j+1} - \zeta^{n+1}_{i,j-1}) \\
&= \left(\frac{F_{ij} + |F_{ij}|}{2}\right)(\zeta^{n+1}_{i,j} - \zeta^{n+1}_{i-1,j}) + \left(\frac{F_{ij} - |F_{ij}|}{2}\right)(\zeta^n_{i+1,j} - \zeta^{n+1}_{i,j}) \\
&\quad + \left(\frac{G_{ij} + |G_{ij}|}{2}\right)(\zeta^{n+1}_{i,j} - \zeta^{n+1}_{i,j-1}) + \left(\frac{G_{ij} - |G_{ij}|}{2}\right)(\zeta^{n+1}_{i,j+1} - \zeta^{n+1}_{i,j}) \\
&\quad + \frac{J^2_{ij}(\zeta^{n+1}_{i,j} - \zeta^n_{i,j})}{\Delta t}
\end{aligned} \tag{19a}$$

and numerical analog of Eq. (18b) was

$$\left(C_{ij}-\frac{E_{ij}}{2}\right)\psi_{i,j-1}+(-2A_{ij}-2C_{ij})\,\psi_{ij}+\left(C_{ij}+\frac{E_{ij}}{2}\right)\psi_{i,j+1}$$

$$=-A_{ij}(\psi_{i+1,j}+\psi_{i-1,j})-\frac{B_{ij}}{4}(\psi_{i+1,j+1}-\psi_{i+1,j-1}-\psi_{i-1,j+1}+\psi_{i-1,j-1})$$

$$-\frac{D_{ij}}{2}(\psi_{i+1,j}-\psi_{i-1,j})+\zeta_{i,j}J_{ij}^2. \tag{19b}$$

The boundary conditions for ψ are given by

$$\psi_{i,1} = 0 \tag{20a}$$

$$\psi_{i,2} = \tfrac{1}{4}\psi_{i,3} \tag{20b}$$

$$\psi_{i,n-1} = \psi_{i,n}-(\Delta n_1)_{i,n}, \tag{20c}$$

where

$$(\Delta n_1)_{i,n}=\sqrt{(x_{i,n_1}-x_{i,n-1})^2+(y_{i,n}-y_{i,n-1})^2}.$$

The value of $\psi_{i,n}$ is a constant, which is updated at every time step. This will be explained later. Equation (20b) is second order accurate, whereas Eq. (20c) is only first order. These numerical representations were chosen for convenience. Equation (20c) worked quite well, because the grid was orthogonal. However, at the inner boundary we used a second order representation to preserve accuracy, because the gradients can be quite large there. The coefficients in Eq. (19a), (19b) are functions of x_ξ, y_ξ, x_η, y_η, J, u, v, evaluated at point ij and Re. The numerical scheme is a line relaxation scheme with updating. It is implicit in the j direction. The convection terms are upwinded and are of first order.

The numerical scheme is made up of the following steps:

Initial conditions are first chosen. They are $\psi_{i,j}=0$ for all i, j. The velocities $u_{i,j}$ and $v_{i,j}$ are set equal to zero except at the outer boundary where they are chosen to give a total tangential velocity of one and a zero normal velocity. The vorticity $\zeta_{i,j}$ is chosen to be zero everywhere. We then proceed as follows:

(1) Evaluate the vorticity at the boundaries, using either the velocity field or the stream function. Here, we use one-sided, second-order, three-point differences if the velocity field is used. Let the vorticity evaluated at the boundary be given by $\tilde{\zeta}_b$. Then the vorticity at the boundary is set equal to

$$\zeta_b^{\text{new}}=\omega\tilde{\zeta}_b+(1-\omega)\,\tilde{\zeta}_b^{\text{old}} \tag{21}$$

where ω is an underrelaxation parameter. Typical values for ω ranged from 0.1 to 0.5. Underrelaxation seems to enhance the stability of the scheme.

(2) Equation (19a) is marched by 1 time step to obtain ζ_{ij}^{n+1}. The method is implicit in the j direction with updating. We obtain a tridiagonal matrix that can be solved using the Thomas algorithm.

(3) The stream function, ψ, at the outer wall is updated by using $\oint d\zeta/\partial n \, ds = 0$. The details of this step will be shown later.

(4) The stream functions, $\psi_{i,2}$ and $\psi_{i,n-1}$, are evaluated using Eqs. (20b) and (20c).

(5) Equation (19b) is iterated (implicitly in j with updating) to convergence

(6) Once the stream function is known, the velocity field is evaluated everywhere in the field.

(7) Check for convergence in the stream function and vorticity fields.

(8) If not converged, go back to step (1).

In step (3) of the numerical scheme, we had mentioned that the value of ψ has to be updated at every time step at the outer boundary. We used the constraint that $\oint \partial\zeta/\partial n \; ds = 0$, where the line integral is along the outer boundary, and $\partial\zeta/\partial n$ is the normal derivative of the vorticity at the outer boundary. This constraint was chosen because we know that for an impermeable moving boundary, with the normal velocity equal to zero and constant tangential velocity, the pressure gradient along the boundary is related to the normal gradient of the vorticity. Also, in this case, $\oint \partial\zeta/\partial n \, ds = 0$ implies $\oint dp = 0$ where the line integral is taken along the outer boundary. For the case of Stokes flow, this would further imply that $\oint dp = 0$ for any closed curve c.

In our problem, we use $\oint \partial\zeta/\partial n \, ds$ to update the stream function at the outer circular wall. We can represent the normal derivative of vorticity by $(\zeta_w - \zeta_{w-1})/(\Delta n_1)$. The vorticity, ζ, at the wall can be expressed in terms of the stream function at the wall. We used $\nabla^2\psi = \zeta$. At the wall we used one-sided differencing, whereas at the interior point we used central differencing. Use was also made of the fact that the grid was orthogonal, and hence some of the terms were dropped. Finally, we can obtain $\partial\zeta/\partial n$ at the outer wall as a function of the stream function, ψ_w, at the outer wall. Thus, setting the numerical integration of $\oint \partial\zeta/\partial n \, ds$ on the outer boundary equal to zero gives us a new value for ψ_w. Note that the finite difference expression should be stable in the sense that if $\oint \partial\xi/\partial n \, ds$ is greater than zero, then Ψ_w should increase, and if it is less than zero, Ψ_w should decrease. To assure this, in the finite difference analog of $\partial\xi/\partial n$ at the outer wall, the stream function at the point adjacent to the wall was replaced by Eq. (20c).

4. *Results and Conclusions*

The problem we chose to solve was the flow in a rotating circular cylinder with a stationary tab with rounded ends. This geometry gave rise to various flow patterns, depending upon the location and size of the tab and the Reynolds number. Figure 7

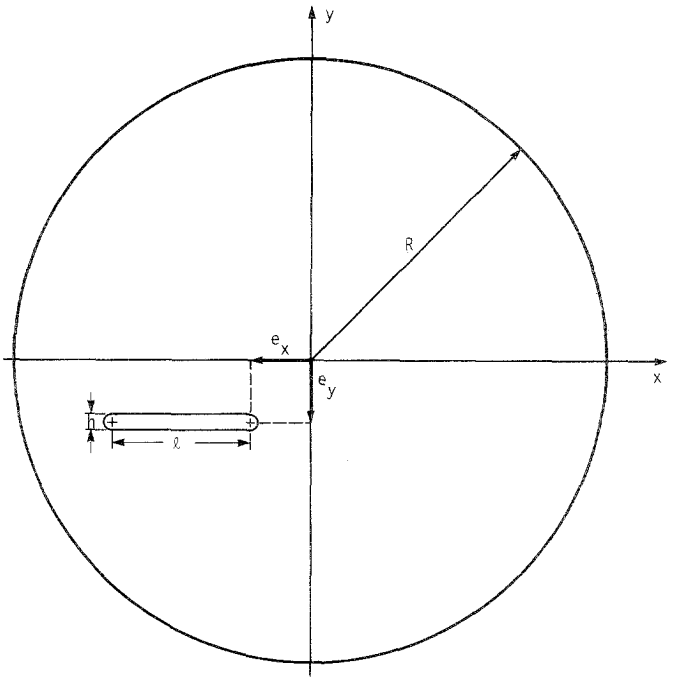

FIG. 7. Type of doubly connected domain in which the Navier–Stokes equations were solved; the outer circular wall rotates.

shows the geometry. In Part I, we generated orthogonal grids in similar, doubly connected domains.

The first case we considered was the flow in an annulus with the inner radius of 0.35 and outer radius of 1, where the outer boundary rotates. This case was chosen because we know the analytic solution. As seen in Fig. 8, the numerical results compare quite favorably with the analytic. Also, the value of the stream function at the outer boundary predicted by the numerical scheme is 0.3503, whereas the analytic value is 0.3534. This case was run at Reynolds number zero. (Note, the analytical solution at higher Reynolds number is the same.) To check the behavior of the scheme at a higher Reynolds number, we ran this same case at different Reynolds numbers. For Reynolds numbers greater than 1, we used the numerical analog for Eq. (13b). For Reynolds number less than 1, we used the numerical analog of Eq. (18a). The underrelaxation factor ω in Eq. (21) was 0.3 for a 31×21 grid. The time step was 5×10^{-3}. At convergence, the maximum change in the stream function from one step to another was of the order of 10^{-6} and that for the vorticity was 10^{-5}.

We found that the results were quite independent of the Reynolds number, as they should be for such a case. We tried a maximum Reynolds number of 500.

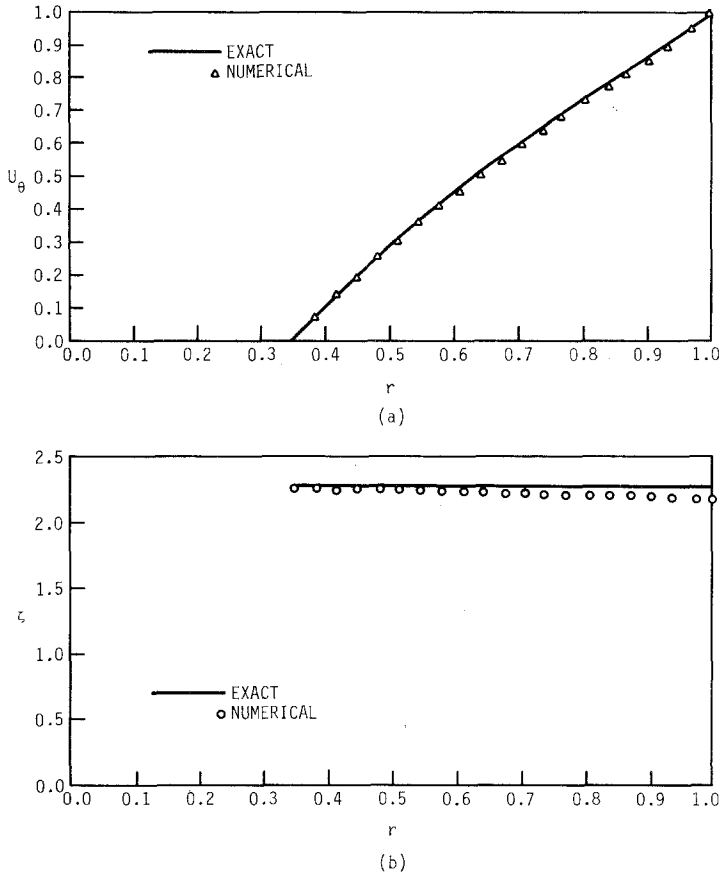

FIG. 8. Comparison of theoretical and numerical results for an annulus with $r_1 = 0.35$ and $r_2 = 1.0$: (a) tangential velocity, U_θ, (b) vorticity, ζ.

Next, we tried doubly connected domains of the type shown in Fig. 7. Figure 9 shows the direction of the velocity vectors for a particular position of the tab. Three different flow patterns are possible for different tab position. These are sketched in Fig. 9. When the tab is sufficiently far from the center, we have a single eddy. As the tab is brought inward, the eddy splits up into two eddies, separated by a saddle point. If we move the tab further toward the center, the saddle point gets attached to the end of the tab, with one eddy on either side of the tab. This was found numerically and verified experimentally for low Reynolds numbers of the order of 0.1. (A brief overview of the experimental techniques used to help verify some of the numerical results obtained is presented in [18].)

To make comparisons with the experiments, numerous numerical runs were made for small changes in the tab position given in terms of e_x and e_y (with e_y kept

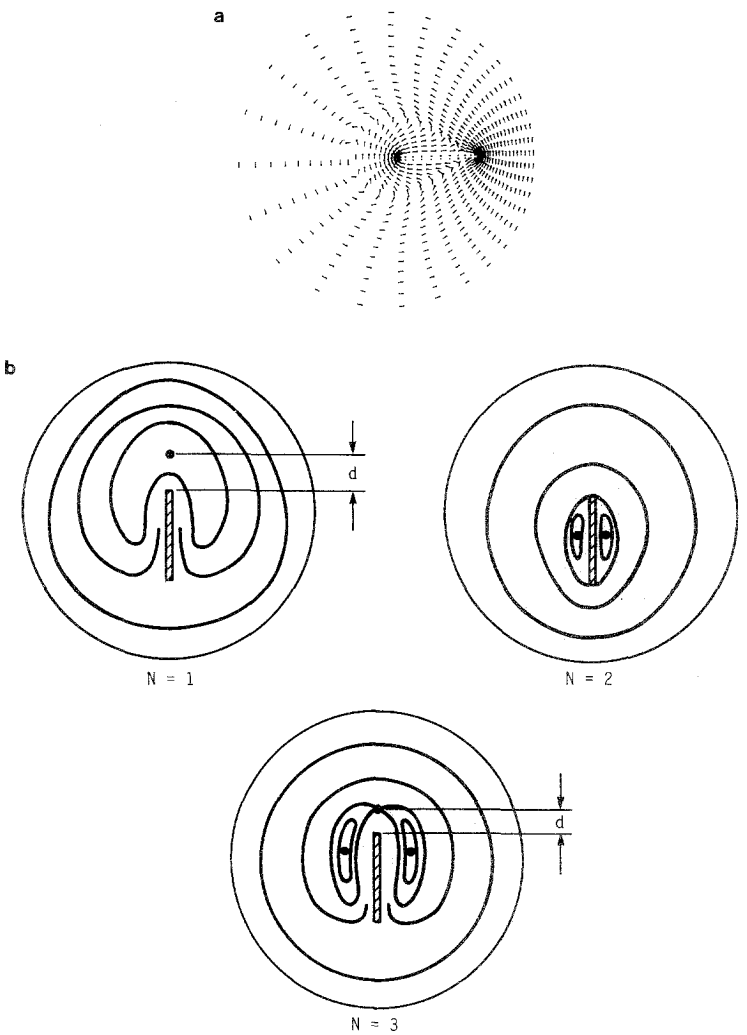

FIG. 9. Stokes flow in a doubly connected domain: (a) flow direction obtained numerically when $e_x = -0.1$, $e_y = 0$, $R = 1.0$, $l = 0.5$, $h = 0.02$, (b) general structure of flow for various values of e_x.

to zero). As we can see from Fig. 10, the numerical results and the experimental results compare very favorably.

Another feature of these flows was the high degree of sensitivity at low Reynolds number to the geometry (see [19]). Experiments showed that if the tab was moved slightly away from the centerline, that is, e_y is not zero but is of the same order of magnitude as the tab thickness, the deviation from symmetry of the flow is considerable. We did try these unsymmetric cases numerically. What we found was that

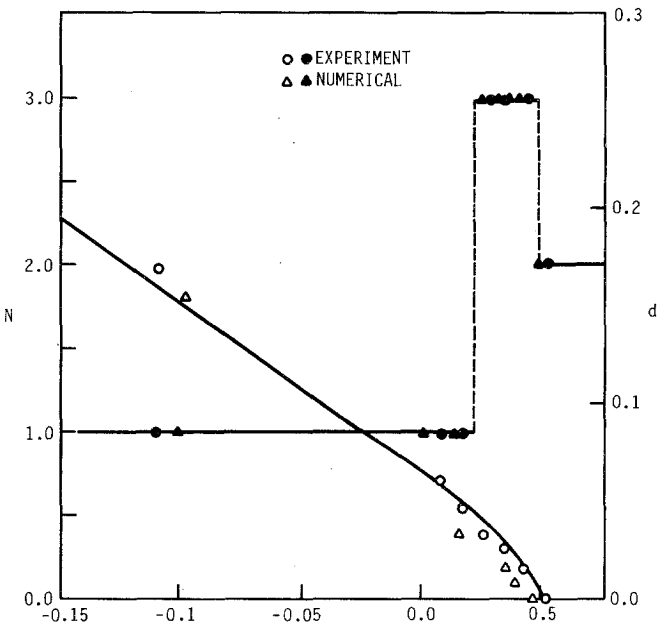

FIG. 10. Structure of flow field as a function of tab position; comparison of numerical and experimental results.

the numerical results also exhibited the same degree of sensitivity to the geometry as the experiments did. This further strengthened the validity of our numerical scheme.

Numerical results for higher Reynolds numbers were also carried out. Figures 11a, b, c show these results for a geometry that was symmetric about the x axis. We note that the flow pattern deviates from the symmetric as the Reynolds number increases. The general structure of the flow is as shown in Fig. 12. With negligible inertia ($\mathrm{Re} \ll 1$), the flow is symmetrical, with one isolated stagnation point (labeled S in Fig. 12) in line with the tab. Inertia effects ($\mathrm{Re} \gg 1$) cause this stagnation point to shift as shown in Fig. 12 as the Reynolds number is increased. The location of the stagnation point, given in terms of α and L/D, where D is the tank diameter, is shown as a function of Re in Fig. 13. The comparison between theory and experiment is excellent.

Another feature of this flow as shown by both the numerical and experimental results is the occurrence of a small recirculating separation zone near the end of the tab (labeled R in Fig. 12) for $\mathrm{Re} \gtrsim 200$. This inertia-induced recirculation is due to the inability of the fluid to turn the relatively sharp corner around the end of the tab when inertia becomes important.

Another feature observed both numerically and experimentally was the position of the rear stagnation point (labeled P in Fig. 12) as a function of Reynolds num-

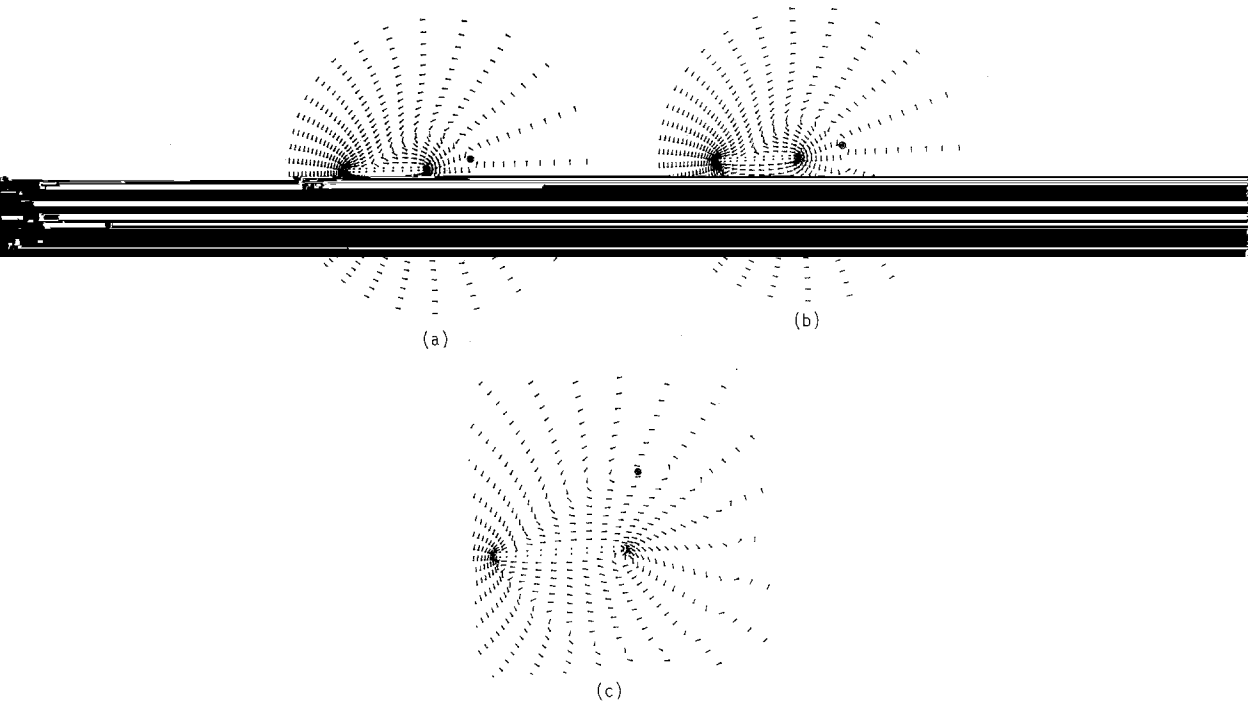

FIG. 11. Navier–Stokes solutions for $e_x = -0.1$, $e_y = 0$, $R = 1.0$, $l = 0.5$, $h = 0.02$ at higher Reynolds number: (a) Re = 10, (b) Re = 50, (c) Re = 500. Approximate stagnation point location denoted by (*).

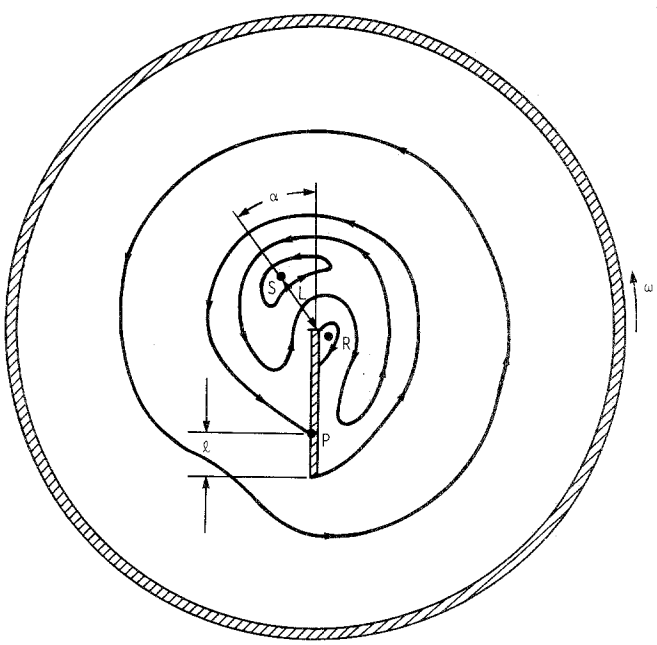

FIG. 12. Nonsymmetry of flow pattern structure when inertia effects are important (Re ≫ 1).

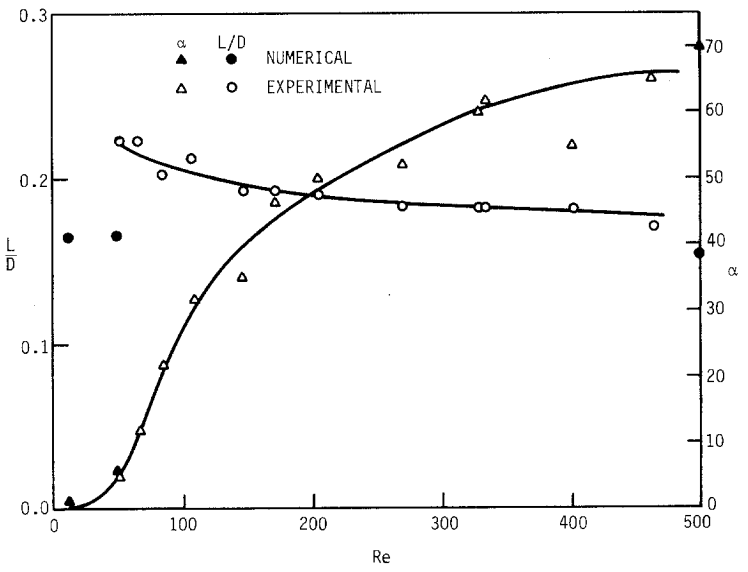

FIG. 13. Comparison of numerical and experimental results for the effect of inertia on the flow structure.

ber. In the limit of $\mathrm{Re} \to 0$ the stagnation point was positioned at the end of the tab; that is, $l = 0$. However, as Re increased, l/D increased until $l/D \cong 0.07$ at $\mathrm{Re} \cong 40$. A further increase in Re caused the stagnation point to move back toward the end of the tab. For example, with $\mathrm{Re} = 500$ we obtained $l/D = 0.02$.

Thus, both the general structure and the specific details for the test flow were very well determined by the numerical technique used. This was true for both the inertia-free ($\mathrm{Re} \to 0$) and the inertia-governed ($\mathrm{Re} \gg 1$) flows. The numerical values for L/D and α were obtained by estimating the stagnation point location within the appropriate grid spacing. This error, when coupled with the experimental error, is of the same size as the difference between the numerical and experimental results of Fig. 13.

5. *Conclusions*

The vorticity stream function method can be successfully used in domains where the value of the stream function on one of the boundaries is not known, but it has to be derived from the solution. Quite good agreements between the numerical and the experimental can be achieved. At low Reynolds numbers, we found no need for underrelaxing the boundary vorticity. However, at higher Reynolds number, we needed underrelaxation for stability.

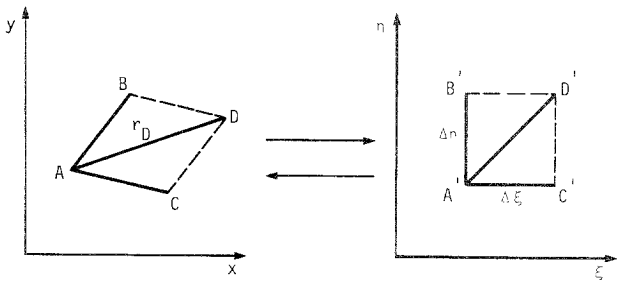

FIG. 14. The mapping of corner points.

APPENDIX

Here we explain what was done near the corner points for the simply connected domain. Consider a point on the boundary in the physical domain, which is mapped to a point on the boundary of the computational domain such that these points are corner points (see Fig. 14). We want BAC to be mapped onto $B'A'C'$.

We use the equation

$$r_D^2 = \left\{ \Delta\xi^2 + \frac{\Delta\eta^2}{K^2} \right\}^{\beta}, \tag{A1}$$

where r_D is the distance of D from A, and K is the value of $K(x, y)$ at A. Since the only unknown is β, we can solve for it. Hence we can obtain

$$r_B^2 = \left(\frac{\Delta\eta}{K} \right)^{2\beta} \tag{A2a}$$

$$r_C^2 = (\Delta\xi)^{2\beta}, \tag{A2b}$$

where r_B and r_C are the distances of points B and C from A along the boundary.

Once points B and C are obtained, a new point D is obtained by applying the Neumann equations in step (5) (of the numerical method to generate the grid) on the two intersecting boundaries and taking the average.

Even though we have no proof for Eq. (A1), we do have some heuristic arguments for it. Since in our scheme for the simply connected domain we have changed the modulus of the domain by changing only $\Delta\eta$ (keeping $\Delta\xi$ fixed), the following is true (for any domain): If $K(x, y) = K_1$ (a constant), and if M_1 is its modulus corresponding to this mapping, and if $K(x, y) = K_2$ (a constant), and if M_2 is the modulus corresponding to this mapping, then

$$\frac{M_1}{M_2} = \frac{K_1}{K_2}. \tag{A3}$$

If $K_2 = 1$, then $M_1/M_2 = K_1$. Also, it follows that all grids generated for a simply connected domain for arbitrary constant $K(x, y)$ look the same. Hence, because of Eq. (A3) and the way we changed our modulus, Eq. (A1) is correct for constant K. However, for a $K(x, y)$ that is not a constant, we still apply Eq. (A1) at the corner. This is based upon the claim that at any point D, such that $K(x, y)$ at D equals K_D (a constant), the mapping generated by $K(x, y)$ and that generated by K_D will be very nearly the same in a small neighborhood of D.

ACKNOWLEDGMENT

This work was supported by the Engineering Research Institute of Iowa State University through funds provided, in part, by National Science Foundation Grant MEA-8204148.

REFERENCES

1. J. F. THOMPSON, F. C. THAMES, AND C. W. MASTIN, *J. Comput. Phys.* **15**, 299 (1974).
2. F. C. THAMES, J. F. THOMPSON, C. N. MASTIN, AND R. L. WALKER, *J. Comput. Phys.* **24**, 245 (1977).
3. J. F. THOMPSON, *AIAA J.* **22**, No. 11, 1505 (1984).
4. B. FORNBERG, *Soc. Ind. Appl. Math. J. Sci. Stat. Comput.* **1**, 386 (1980).
5. T. K. HUNG AND T. D. BROWN, *J. Comput. Phys.* **23**, 343 (1977).
6. S. B. POPE, *J. Comput. Phys.* **26**, 197 (1978).
7. C. D. MOBLEY AND R. J. STEWART, *J. Comput. Phys.* **34**, 124 (1980).
8. D. GAIER, in *Lecture Notes in Mathematics*, No. 399 (Springer-Verlag, Berlin, 1974).
9. L. V. AHLFORS, *Lectures in Quasiconformal Mappings* (Van Nostrand, Princeton, NJ, 1966).
10. G. RYSKIN AND L. G. LEAL, *J. Comput. Phys.* **50**, 71 (1983).
11. E. D. CHIKHLIWALA AND Y. C. YORTOS, *J. Comput. Phys.* **57**, 391 (1985).
12. P. J. ROACHE, *Computational Flow Dynamics* (Hermosa, Albuquerque, NM, 1972).
13. D. C. THOMAS AND A. A. SZEWCZYK, Tech. Report 66-14, Heat Transfer and Fluid Mechanics Lab., Department of Mechanical Engineering, University of Notre Dame, July 1966 (unpublished).
14. S. I. CHENG, *AIAA J.* **8**, No. 12, 2115 (1970).
15. W. R. BREILY, Report J110614-1, United Aircraft Research Laboratories, East Hartford, CN, 1970 (unpublished).
16. C. E. PEARSON, *J. Fluid Mech.* **21**, No. 4, 611 (1965).
17. L. C. WOODS, *Aeronaut. Q.* **5**, No. 3, 176 (1954).
18. B. R. MUNSON AND L. D. STURGES, *Phys. Fluids* **26**, No. 5, 1173 (1983).
19. B. R. MUNSON AND B. GUSTAFSON, *Exp. Fluids* **3**, No. 257 (1985).