J. Fluid Mech. (1965), *vol.* 21, *part* 4, *pp.* 611–622
Printed in Great Britain

A computational method for viscous flow problems

By CARL E. PEARSON
Sperry Rand Research Center, Sudbury, Massachusetts

(Received 21 July 1964)

A digital computer method for solving certain problems involving two-dimensional incompressible viscous flow is described. The time-dependent case is treated; the mathematical problem is thus that of solving a non-linear fourth-order partial differential equation in three variables. The choice of difference equations, of relaxation procedure, the kind of approximation to boundary conditions, and the resulting computational stability, speed, and accuracy are considered. Most experience so far has been for a rectangular region for which boundary velocities are prescribed as certain functions of time; an example of one such problem showing vortex formation and break-up is given.

1. Introduction

We describe here a computational method which seems to be effective for time-dependent two-dimensional viscous flow problems. The method itself is outlined in § 3, and some remarks concerning details of procedure are given in § 4. § 5 presents some computational results.

Mathematically, the problem is that of solving numerically a non-linear fourth-order partial differential equation in three independent variables. It is known that, even without the added complications of time variation or non-linearity, numerical techniques for fourth-order equations tend to require substantial computer time. The presence of non-linear terms may not only accentuate the computer time problem, but may also tend to induce computational instabilities. Moreover, the most natural boundary conditions to impose are usually those where boundary velocities are prescribed; this is equivalent to specifying the first derivatives of the unknown function on the boundary, and for a fourth-order equation such boundary conditions can be difficult to treat properly.

As might be expected from their potential importance, viscous flow computations have received considerable attention in the literature (Fromm & Harlow 1963; Dix 1963; Hellums & Churchill 1961; Wilkes 1963 and Thompson 1961 are representative). In general, it appears that there have been difficulties with one or another of computational speed, stability, accuracy, or the ability to handle correct boundary conditions. In a problem of this general difficulty, it seems worth while to begin with the simplest possible problem possessing realistic boundary conditions. For this reason, most of our work to date has concerned the solution of the time-dependent Navier–Stokes equations in a rectangular region, on the edge of which the tangential and normal velocities are prescribed for all time. Attention has been concentrated on the basic method itself; extensions

to permit curved boundaries, variable mesh spacings, multiple connectivity, or the presence of thermal or electromagnetic terms have yet to be made. Most experimentation with the method has been in connexion with a test problem whose exact solution is known; accuracies of a fraction of a percent are obtained, even with time steps of up to 50 times those imposed by conventional stability limitations. The general method has also been applied to two problems whose solutions are not known. One of these concerns the injection of a fluid into a rectangular region; the results can here be verified by repeating the calculation with different space or time meshes. The second problem is a time-dependent version of a rotating disk problem, and is described in the following paper (Pearson 1965).

2. Equations and boundary conditions

In terms of the usual Cartesian co-ordinates (x, y) and time t, the equations relating pressure p, and velocity components (u, v) for two-dimensional incompressible flow are

$$\rho(u_t + u_x u + u_y v) = -p_x + \mu(u_{xx} + u_{yy}). \tag{1}$$

$$\rho(v_t + v_x u + v_y v) = -p_y + \mu(v_{xx} + v_{yy}), \tag{2}$$

$$u_x + v_y = 0, \tag{3}$$

where ρ is density and μ is viscosity; subscripts denote partial differentiation. Equation (3) is equivalent to

$$u = \psi_y, \quad v = -\psi_x, \tag{4}$$

where ψ is the stream function. The vorticity ξ is defined by

$$\xi = -\tfrac{1}{2}(u_y - v_x) = -\tfrac{1}{2}\Delta\psi \tag{5}$$

and equations (1) and (2) may then be differentiated and combined to give

$$d\xi/dt = \xi_t + \xi_x\psi_y - \xi_y\psi_x = \nu\Delta\xi, \tag{6}$$

where $\nu = \mu/\rho$ is the kinematic viscosity. Equation (6), the vorticity transport equation, can be thought of as the condition that a pressure function p exist, with derivatives obtainable from (1) and (2). An equation for p may be obtained from (1) and (2):

$$\Delta p = 2\rho[\psi_{xx}\psi_{yy} - (\psi_{xy})^2]. \tag{7}$$

In a multiply-connected region, ψ as defined by (4) may be multiple-valued. The most usual case, however, is that in which the multiple-connectedness is caused by the insertion of an obstacle in the flow; since there is no net flow of fluid through the obstacle, $\oint d\psi = 0$, so that in such a case ψ remains single-valued.

On a fixed solid boundary, we must have $u = v = 0$, or equivalently,

$$\partial\psi/\partial s = \partial\psi/\partial n = 0, \tag{8}$$

where these two derivatives are in the tangential and normal directions respectively. It follows easily from (1), (2) and (5) that on such a boundary

$$\partial p/\partial n = 2\mu\,\partial\xi/\partial s, \quad \partial p/\partial s = -2\mu\,\partial\xi/\partial n. \tag{9}$$

It has been mentioned already that (6) is the condition that a pressure function p, with derivatives given by (1) and (2), should exist; by Stokes's theorem this condition is equivalent to the requirement that

$$p(\mathrm{B}) - p(\mathrm{A}) = \int_{\mathrm{A}}^{\mathrm{B}} (p_x\,dx + p_y\,dy) \tag{10}$$

be independent of the path joining A to B. However, this holds only for paths linking A to B that may be continuously deformed into one another, so that—even if (6) is satisfied—p as determined from (10) could be multiple-valued for paths enclosing an obstacle. Since p is a single-valued quantity, we must include in the formation of a problem the explicit requirement that the integral (10) be zero for any circuit enclosing an obstacle.

3. Basic method

In using a finite-difference method, the object is to calculate the values of ψ and ξ at the intersections of the mesh lines shown in figure 1, for each discrete time step. Denote the mesh spacing by h, and the length of time step by δt; denote the values of ψ and ξ at $x = ih$, $y = jh$, $t = n\,\delta t$ by $\psi^n(i,j)$ and $\xi^n(i,j)$ respectively. In general, we use a finite difference analogue of (6) to compute $\xi^{n+1}(i,j)$ from known values of $\xi^n(i,j)$ and $\psi^n(i,j)$, for all i,j; in turn, values of $\psi^{n+1}(i,j)$ may then be obtained from the finite difference analogue of (5).

The usual finite difference expression for $\Delta\xi$ at time $n\,\delta t$ is

$$\Delta_h \xi^n(i,j) = h^{-2}[\xi^n(i+1,j) + \xi^n(i-1,j) + \xi^n(i,j+1) + \xi^n(i,j-1) - 4\xi^n(i,j)]. \tag{11}$$

The finite difference form of (6) could be written so as to provide an explicit formula for $\xi^{n+1}(i,j)$ in terms of $\xi^n(i,j)$, $\Delta_h\xi^n(i,j)$, and certain first derivative products; however, $\Delta_h\xi^n(i,j)$ cannot be calculated for those points marked B in figure 1, since equation (11) cannot be applied there. (The boundary conditions prescribe ψ and $\partial\psi/\partial n$ on the boundary, but not ξ, so we do not know ξ^n on points A.) Nevertheless, since the values of ψ^{n+1} at points A and B can be determined from the prescribed boundary conditions, a knowledge of ξ^{n+1} in the innermost region C is enough, since we can then solve (5) as a Poisson equation in region C, using known values of ψ^{n+1} on points B; thus ψ^{n+1} in C could be determined. With all ψ^{n+1} now known, ξ^{n+1} on points B can be determined from (5) (using the values of ψ^{n+1} at each surrounding point); these values of ξ^{n+1} on points B are, of course, necessary for the next time step, in order to compute $\Delta_h\xi^{n+1}$ at such points as those marked D.

Thus the two differential equation problems (5) and (6) are coupled together by the values of ξ on points B. However, the process so far described would be unsatisfactory in several respects; it is computationally unstable for large time steps, is not particularly accurate even for small time steps, and moreover, computes ξ^{n+1} for points B and C by different procedures. We consider next the matter of calculating ξ^{n+1} at points in region C in a more accurate manner. If the term ξ_t in (6) is replaced by $[\xi^{n+1}(i,j) - \xi^n(i,j)]/\delta t$, then it is clearly more accurate to use values of ξ and ψ in the rest of the equation as determined for time $(n+\frac{1}{2})\,\delta t$, rather than for time $n\,\delta t$. Thus we should, for example, replace

$\Delta_h \xi^n(i,j)$ by $\frac{1}{2}[\Delta_h \xi^n(i,j) + \Delta_h \xi^{n+1}(i,j)]$. However, the calculation of $\Delta_h \xi^{n+1}(i,j)$ requires a knowledge of $\xi^{n+1}(i,j)$, so that the use of this kind of averaging process in the finite difference equations results in the equations becoming implicit, rather than explicit; we obtain a set of algebraic equations with as many unknowns as there are mesh points. Nevertheless, since the explicit process described before is computationally stable only for very small time steps δt, and even then only if the non-linear terms in (6) are small, and since the implicit method does not have this defect, this change to an implicit process is of crucial importance.

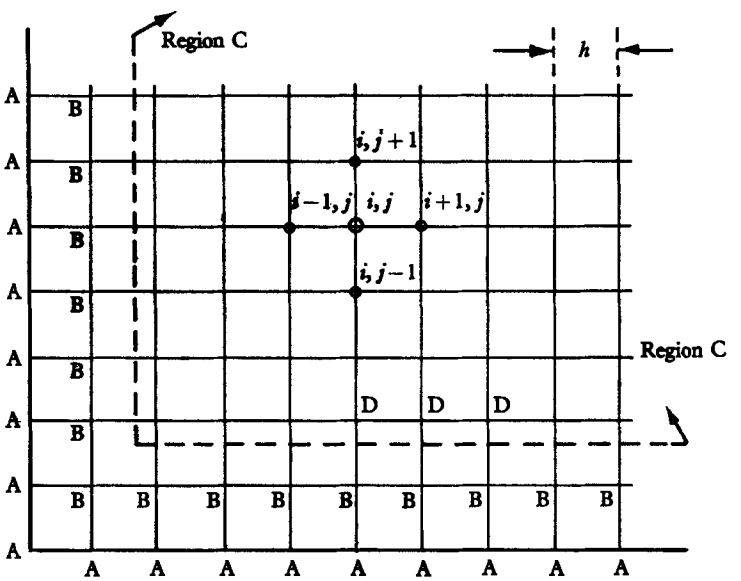

FIGURE 1. Mesh geometry.

The implicit equation is

$$
\begin{aligned}
[\xi^{n+1}(i,j) - \xi^n(i,j)]/\delta t = {} & (\nu/h^2)\,[\tfrac{1}{2}\Delta_h \xi^n(i,j) + \tfrac{1}{2}\Delta_h \xi^{n+1}(i,j)] \\
& + (1/16h^2)\,[\xi^{n+1}(i,j+1) + \xi^n(i,j+1) - \xi^{n+1}(i,j-1) - \xi^n(i,j-1)] \\
& \quad \times [\psi^{n+1}(i+1,j) + \psi^n(i+1,j) - \psi^{n+1}(i-1,j) - \psi^n(i-1,j)] \\
& - (1/16h^2)\,[\xi^{n+1}(i+1,j) + \xi^n(i+1,j) - \xi^{n+1}(i-1,j) - \xi^n(i-1,j)] \\
& \quad \times [\psi^{n+1}(i,j+1) + \psi^n(i,j+1) - \psi^{n+1}(i,j-1) - \psi^n(i,j-1)]. \qquad (12)
\end{aligned}
$$

It will be observed that centred space differences are used. Equation (12) is solved by successive optimal over-relaxation sweeps (see Forsythe & Wasow 1960); for a rectangular region, the optimal over-relaxation parameter can be explicitly calculated.

Again, this process cannot include points B in figure 1. Moreover, since values of ξ^{n+1} at points B occur when (12) is applied to points D, we must use guessed values for ξ^{n+1} at points B, which are corrected after values of ψ^{n+1} have been determined by use of (5). Thus an iterative process is required, in which the 'hinge' values of ξ^{n+1} at points B are successively corrected. This same iteration process allows us to generate successively improved values for ψ^{n+1} for use in (12).

From (5), we obtain

$$\xi^{n+1}(i,j) = -(1/2h^2)\,[\Delta_h \psi^{n+1}(i,j)], \tag{13}$$

which can be solved for $\psi^{n+1}(i,j)$, again by successive optimal over-relaxation, from known values of ξ^{n+1} in region C, and from given boundary values of ψ^{n+1} on points B.

The iterative process just described, in which (12) and (13) are solved repeatedly in sequence until values of ξ^{n+1} and ψ^{n+1} no longer change, is very sensitive to the values of ξ^{n+1} on points B. To avoid instabilities, it is necessary to introduce a smoothing parameter, so that the 'new' values of ξ^{n+1} at points B obtained in each iteration sweep are taken to be weighted averages of the 'old' values and the values obtained by applying (13) to points B. Similarly, when the non-linear terms in (12) are large, the ψ^{n+1} coupling evident in the last two terms can lead to instability, and for that reason the new internal values of ξ^{n+1} as obtained by solving (12) are also weighted with the previous values.

In determining the values of ψ^{n+1} on points B, as determined from prescribed values of ψ and $\partial\psi/\partial n$ at time $(n+1)\,\delta t$ on the boundary, it is necessary to use a formula accurate to second-order terms (since the second-order ψ terms are not negligible in the vorticity calculations). Thus, for example, at time $t = (n+1)\,\delta t$,

$$\psi(\mathrm{B}) = \psi(\mathrm{A}) + h\,\partial\psi(\mathrm{A})/\partial n + \tfrac{1}{2}h^2\,\partial^2\psi(\mathrm{A})/\partial n^2 + \ldots. \tag{14}$$

Here $\psi(\mathrm{A})$ and $\partial\psi(\mathrm{A})/\partial n$ are prescribed, but $\partial^2\psi(\mathrm{A})/\partial n^2$ must be determined from $\xi^{n+1}(\mathrm{A})$, which in turn can be obtained by extrapolation from $\xi^{n+1}(\mathrm{B})$ and $\xi^{n+1}(\mathrm{D})$. It is clear that (14) will then also be an ingredient in the iterative process.

Finally, we may remark that (12) is very rapidly solved by the over-relaxation process when the non-linear terms are small (because the main diagonal terms are strongly dominant); if the non-linear terms are large, equation (12) is unsatisfactory and must be replaced by an implicit alternating direction method. The method chosen is based on that devised by Peaceman & Rachford (1955) for diffusion problems. Each time-step is now covered in two half-steps:

$$\begin{aligned}
[\xi^{n+\frac{1}{2}}(i,j) - \xi^{n}(i,j)]/\tfrac{1}{2}\,\delta t &= (\nu/h^2)\,[\xi^{n+\frac{1}{2}}(i+1,j) + \xi^{n+\frac{1}{2}}(i-1,j) - 2\xi^{n+\frac{1}{2}}(i,j) \\
&\qquad + \xi^{n}(i,j+1) + \xi^{n}(i,j-1) - 2\xi^{n}(i,j)] \\
&+ (1/4h^2)\,[\xi^{n}(i,j+1) - \xi^{n}(i,j-1)] \\
&\qquad \times [\tfrac{1}{4}\psi^{n+1}(i+1,j) + \tfrac{3}{4}\psi^{n}(i+1,j) - \tfrac{1}{4}\psi^{n+1}(i-1,j) - \tfrac{3}{4}\psi^{n}(i-1,j)] \\
&+ (1/4h^2)\,[\xi^{n+\frac{1}{2}}(i+1,j) - \xi^{n+\frac{1}{2}}(i-1,j)] \\
&\qquad \times [\tfrac{1}{4}\psi^{n+1}(i,j+1) + \tfrac{3}{4}\psi^{n}(i,j+1) - \tfrac{1}{4}\psi^{n+1}(i,j-1) - \tfrac{3}{4}\psi^{n}(i,j-1)], \qquad (15)
\end{aligned}$$

$$\begin{aligned}
[\xi^{n+1}(i,j) - \xi^{n+\frac{1}{2}}(i,j)]/\tfrac{1}{2}\,\delta t &= (\nu/h^2)\,[\xi^{n+\frac{1}{2}}(i+1,j) + \xi^{n+\frac{1}{2}}(i-1,j) - 2\xi^{n+\frac{1}{2}}(i,j) \\
&\qquad + \xi^{n+1}(i,j+1) + \xi^{n+1}(i,j-1) - 2\xi^{n+1}(i,j)] \\
&+ (1/4h^2)\,[\xi^{n+1}(i,j+1) - \xi^{n+1}(i,j-1)] \\
&\qquad \times [\tfrac{3}{4}\psi^{n+1}(i+1,j) + \tfrac{1}{4}\psi^{n}(i+1,j) - \tfrac{3}{4}\psi^{n+1}(i-1,j) - \tfrac{1}{4}\psi^{n}(i-1,j)] \\
&+ (1/4h^2)\,[\xi^{n+\frac{1}{2}}(i+1,j) - \xi^{n+\frac{1}{2}}(i-1,j)] \\
&\qquad \times [\tfrac{3}{4}\psi^{n+1}(i,j+1) + \tfrac{1}{4}\psi^{n}(i,j+1) - \tfrac{3}{4}\psi^{n+1}(i,j-1) - \tfrac{1}{4}\psi^{n}(i,j-1)]. \qquad (16)
\end{aligned}$$

Each coefficient matrix is tri-diagonal, so that one double sweep of Gaussian elimination solves each equation set.

We now list the steps in the computational procedure required to go from time $n\,\delta t$ to time $(n+1)\,\delta t$. It is assumed that values of ξ^n and ψ^n are known, and it is desired to find values of ξ^{n+1} and ψ^{n+1}:

(*a*) Obtain a first approximation for ξ^{n+1} and ψ^{n+1} at each required interior mesh point by time extrapolation from previously computed values of ξ and ψ. Insert values of ψ^{n+1} on points A, as given boundary conditions.

(*b*) Solve (12) (or (15) and (16)) to give ξ^{n+1}(C).

(*c*) Use (14), with specified boundary values of $\partial\psi/\partial n$ at time $(n+1)\,\delta t$, to compute approximate values of ψ^{n+1}(B).

(*d*) Using these values of ψ^{n+1}(B) on the boundary of region C, solve (13) to give ψ^{n+1}(C).

(*e*) Use (13) to obtain revised values for ξ^{n+1}(B), but use a 'smoothed' result, as obtained by weighting with previous values.

(*f*) Repeat step (*b*), to give revised values of ξ^{n+1}(C). Again, weight these revised values with the previous values, so as to produce 'smoothed' revised values.

(*g*) Repeat the whole process iteratively until convergence has resulted. The time step is then complete.

4. Remarks

(*a*) If an explicit equation were used instead of (12), the usual stability criterion would require $\nu\,\delta t < \frac{1}{4}h^2$ (if the non-linear terms had no effect). To minimize the effect of the non-linear terms, right-handed or left-handed space derivatives have been suggested (Hellums & Churchill 1961); not only does this result in a loss in accuracy, but the requirement of taking very small time steps is not avoided.

In general, there are three considerations governing the choice of δt; these are computational stability, accuracy, and compatibility with boundary data. By the last requirement is meant that δt must be reasonably small compared to the time scale over which boundary data is changing.

A difference equation scheme which has sometimes been suggested (see, for example, Fromm & Harlow 1963) is that of DuFort & Frankel (1953). This consists in replacing (12) by

$$\begin{aligned}[\xi^{n+1}(i,j)-\xi^{n-1}(i,j)]/2\,\delta t = (\nu/h^2)\,[&\xi^n(i,j+1)+\xi^n(i,j-1)\\ &+\xi^n(i+1,j)+\xi^n(i-1,j)-2\xi^{n+1}(i,j)-2\xi^{n-1}(i,j)]\\ &+\text{other terms.}\end{aligned} \tag{17}$$

Equation (17) is essentially in explicit form, and moreover has better stability properties than the usual explicit analogue of (12). However, as recognized by DuFort & Frankel themselves, a truncation-error analysis of (17) shows that the accuracy of the difference approximation is much less than that of either the explicit equation or of (12). In fact, it usually turns out that, for reasons of accuracy, one must impose a δt limitation similar to that required for stability

of the explicit equation. These expectations have been verified by computational experimentation with the test problem to be described; in a typical trial (with $\nu\,\delta t/h^2 = 4$), the DuFort–Frankel equation gave errors averaging 2·3 %, whereas the implicit equation gave errors averaging 0·012 %.

In equation (12), there is an equal contribution of terms in n and in $n+1$ on the right-hand side. Experiments were run for various cases of unequal contributions, but no advantages were found to ensue.

If the non-linear terms in (6) were, instead, linear in ξ with constant coefficients, then the usual kind of stability analysis shows that (12) is unconditionally stable for all δt. Experience indicates that this property seems to carry over to the non-linear case; moreover, (12) gives high accuracy even with large δt, so that, in many problems, the use of (12) can result in substantial savings in computer time.

$\nu\,\delta t/h^2$	Q	I
51·2	0·80	No convergence
51·2	0·85	23
51·2	0·90	34
51·2	0·95	50
12·8	0·80	No convergence
12·8	0·85	13
12·8	0·90	16
12·8	0·95	40
3·2	0·80	No convergence
3·2	0·85	10
3·2	0·90	16
3·2	0·95	28

TABLE 1.

(*b*) Analysis shows that (15) and (16) are also unconditionally stable (if the non-linear terms are approximated locally by linear terms). A discretization error analysis shows that (15) and (16) are remarkably close to (12) in accuracy, and again this was confirmed experimentally for the problem to be described in § 5. For situations where the non-linear terms are large, (15) and (16) may be somewhat less accurate than (12), but, for large non-linear terms, (12) requires many relaxation sweeps.

(*c*) The optimal over-relaxation process for (12) was so fast (at least for the case of small non-linear terms) that it seemed worth while to try an optimal over-relaxation process for the fourth-order equation in ψ obtained by combining (5) and (6). The optimal over-relaxation parameter was determined experimentally. The results were very unsatisfactory; it apparently takes an order of magnitude more computer time to solve the single equation by optimal over-relaxation than it does to solve the coupled equation set.

(*d*) The result of the iteration process is that all equations are satisfied exactly at all points; thus the fact that different methods are used for points B and C is of no significance. The effect of the smoothing parameter at the boundary is interesting; for a case in which the non-linear terms were small, so that no 'internal' smoothing was required, the results in table 1 were obtained. Here Q

is the weighting factor (proportion of old value), and I is the number of iterations required for convergence. The calculations were carried out with a total of 400 mesh points.

When internal smoothing was also used (and if the non-linear terms were large, it was essential to do so), the convergence speed was found to increase even for those cases when there was already convergence without its use. In a typical case, the requisite number of iterations was cut by a third.

(*e*) If the last term in (14) was not included, experience showed that errors in ξ near the boundary of up to 50 % could result. Incidentally, the inclusion of this term means that the boundary really is at points A, and not at some position 'in-between' points A and B.

It is possible to hinge together the two equations, (5) and (6), at points A rather than at points B. To do this, values of ξ must be obtained at points A. Since

$$\psi(\mathrm{B}) = \psi(\mathrm{A}) + h\,\partial\psi(\mathrm{A})/\partial n + \tfrac{1}{2}h^2\,\partial^2\psi(\mathrm{A})/\partial n^2 + \tfrac{1}{6}h^3\,\partial^3\psi(\mathrm{A})/\partial n^3$$

and

$$\psi(\mathrm{D}) = \psi(\mathrm{A}) + 2h\,\partial\psi(\mathrm{A})/\partial n + \tfrac{1}{2}(2h)^2\,\partial^2\psi(\mathrm{A})/\partial n^2 + \tfrac{1}{6}(2h)^3\,\partial^3\psi(\mathrm{A})/\partial n^3 \quad (18)$$

and since $\psi(\mathrm{A})$ and $\partial\psi(\mathrm{A})/\partial n$ are specified as boundary conditions, $\partial^2\psi/\partial n^2$ can be obtained by elimination between these two equations. The fact that the second derivative of ψ along the boundary is known then permits ξ at A to be computed. This type of boundary connexion between equations (5) and (6) may be particularly useful for regions with re-entrant corners; preliminary experimentation by Dr R. Esch with the present method, but with (14) replaced by (18), indicates that accuracy and convergence speed are comparable. Equation (18) has been used by Wilkes (1963) in a viscous heat-conduction problem; (15) and (16) were also used by Wilkes, but no iteration on boundary vorticities was carried out.

If the boundary conditions involve a prescription of stream function and vorticity rather than the normal derivative of the stream function, then equations (5) and (6) are largely decoupled and the problem is much easier. This is the method used by Fromm & Harlow (1963); boundary values of ξ were given approximate values based on their values as obtained for a simpler problem. Fromm & Harlow used the DuFort–Frankel equation; their region was multiply connected, but apparently equation (10) was not satisfied for each of the two possible circuits.

(*f*) Further details of the method, and calculations of stability, etc., will be found in Pearson (1964).

5. Results

An exact solution of equations (5) and (6) is given by

$$\psi = \exp(-2\pi^2 t)\cos\pi x\cos\pi y$$

with $\nu = 1$. The region of interest is the quarter-wave region $0 \leqslant x \leqslant \frac{1}{2}$, $0 \leqslant y \leqslant \frac{1}{2}$. Most of the computer experimentation was based on this problem, where exact and computed results could be compared. The non-linear terms vanish for the exact solution but do not quite vanish in the difference equation approximations; thus, any instability caused by their presence would be

detected. The method described in § 3 gave excellent results for this problem. A typical case is that in which there are 400 mesh points, with $\delta t/h^2 = 3{\cdot}2$ (which is 8 times the stability limit for the explicit equation, or the equivalent accuracy limit for the DuFort–Frankel method); table 2 compares calculated and exact values of ψ and ξ at a point 3 mesh spaces in from each side, at each of a number of different times. Here I is the number of iterations. The boundary conditions consisted in a specification of exact values of ψ and $\partial\psi/\partial n$.

t	ψ (calc.)	ψ (exact)	ξ (calc.)	ξ (exact)	I
0	0·97552824	0·97552824	9·6231221	9·6280778	—
0·002	0·93773540	0·93776621	9·2411754	9·2553814	23
0·004	0·90142716	0·90146594	8·8787761	8·8971122	20
0·006	0·86653421	0·86657084	8·5378473	8·5527112	18
0·008	0·83299073	0·83302648	8·2065818	8·2216416	13
0·010	0·80074655	0·80078062	7·8892688	7·9033878	14
0·012	0·76974986	0·76978296	7·5835788	7·5974532	5
0·014	0·73995353	0·73998519	7·2902472	7·3033611	4
0·016	0·71131037	0·71134090	6·5575255	6·5670896	5
0·018	0·68377611	0·68380538	6·7367162	6·7488886	4
0·020	0·65730756	0·65733575	6·4758809	6·4876438	4
0·022	0·63186368	0·63189074	6·2823232	6·2947859	4
0·034	0·49859951	0·49862087	4·9122772	4·9211907	4
0·036	0·47929906	0·47931961	4·7221233	4·7306949	4
0·054	0·33596905	0·33598344	3·3100098	3·3160236	3
0·056	0·32296392	0·32297777	3·1818811	3·1876627	3

TABLE 2.

The running time, through a time equal to the time constant of the problem, was about $\frac{1}{3}$ hour on an IBM 7090 computer.

A fluid-injection problem, whose solution was not known, was also analysed. The region of interest was here $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$, and the boundary conditions were taken as $\psi = \psi_n = 0$ on all sides other than the side $y = 0$. On $y = 0$, the boundary condition chosen was

$$\psi = 10(\cos 2\pi x - 1)(1 - e^{-100t}), \quad \psi_n = 0. \tag{19}$$

Again, non-dimensional variables were used, so that $\nu = 1$. Equation (19) corresponds to fluid being injected in the interval $0 \leqslant x \leqslant \frac{1}{2}$ and withdrawn in the interval $\frac{1}{2} \leqslant x \leqslant 1$, directions of both injection and extraction being perpendicular to the side $y = 0$. The time-step chosen was $\delta t = 0{\cdot}001$, and it was found that the choices $\delta t = 0{\cdot}0005$, $\delta t = 0{\cdot}002$, did not appreciably alter the results, nor the number of iterations required per time step. The space interval was $h = 0{\cdot}025$, corresponding to a total of 1681 mesh points. Table 3 shows values of ψ obtained, with the three different choices for δt, at $t = 0{\cdot}002$, and on the first 10 mesh points on the row $y = 0{\cdot}25$.

With $\delta t = 0{\cdot}001$, the first time-step required 15 iterations for convergence, the fourth 15 iterations, and the eighth 10 iterations. The motion was essentially steady-state by the 36th time-step; here 3 iterations were required. The boundary

smoothing parameter used was 0·85; no internal smoothing was required. The final steady-state stream-function plot is shown in figure 2. The closed contour represents reverse flow.

$\delta t = 0{\cdot}0005$	$\delta t = 0{\cdot}001$	$\delta t = 0{\cdot}002$
0·0	0·0	0·0
−0·031	−0·032	−0·034
−0·105	−0·106	−0·111
−0·203	−0·204	−0·209
−0·311	−0·312	−0·317
−0·425	−0·426	−0·429
−0·541	−0·541	−0·543
−0·657	−0·657	−0·657
−0·772	−0·772	−0·770
−0·885	−0·884	−0·881
−0·994	−0·993	−0·989
−1·098	−1·098	−1·092

TABLE 3. Comparison of ψ-values for various time-step sizes

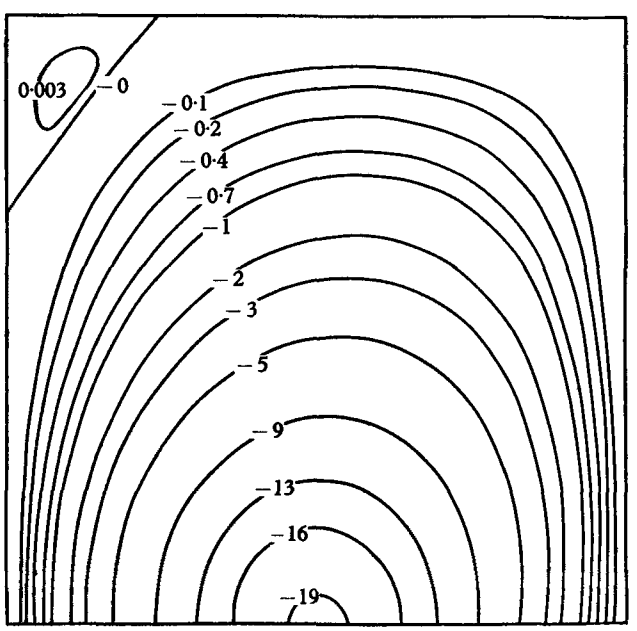

FIGURE 2. Stream-function contours for re-entrant viscous flow. Amplitude = 10; time = 0·036.

With the amplitude in equation (19) increased by a factor of 10, the final steady-state flow pattern shown in figure 3 was obtained.

With the amplitude increased by a further factor of 10, the patterns shown in figures 4 and 5 were obtained at times 0·010 and 0·022. It was now essential to use internal smoothing; otherwise, computational instability would set in at about $t = 0{\cdot}01$. Figures 4 and 5 were obtained with 400 mesh points, so that

figure 5 in particular is probably not too accurate, since the scale of motion is now approaching the mesh size. Nevertheless, the way in which the reverse flow region of figure 4 has been pinched off into two halves so as to result in the configuration of figure 5 is interesting. Although the boundary conditions are symmetric, the differential equations are not (because of the non-linear terms),

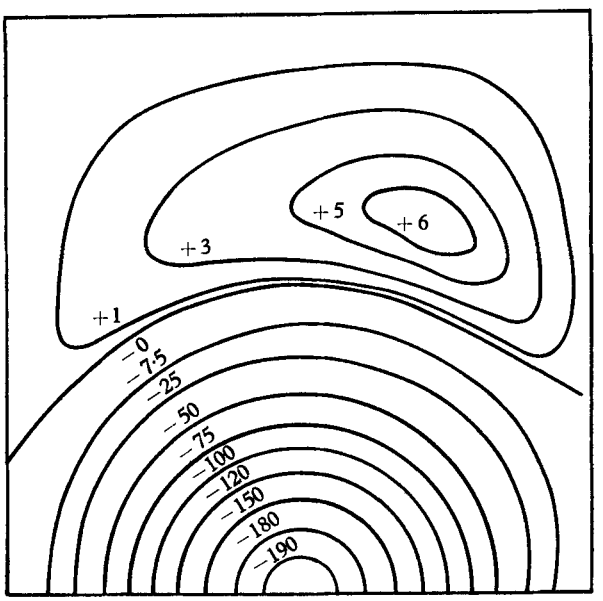

FIGURE 3. Stream-function contours for re-entrant viscous flow.
Amplitude = 100; time = 0·050.

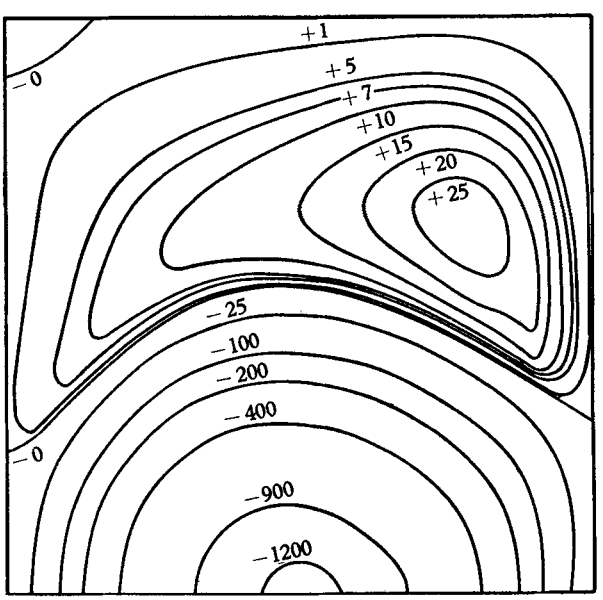

FIGURE 4. Stream-function contours for re-entrant viscous flow.
Amplitude = 1000; time = 0·010.

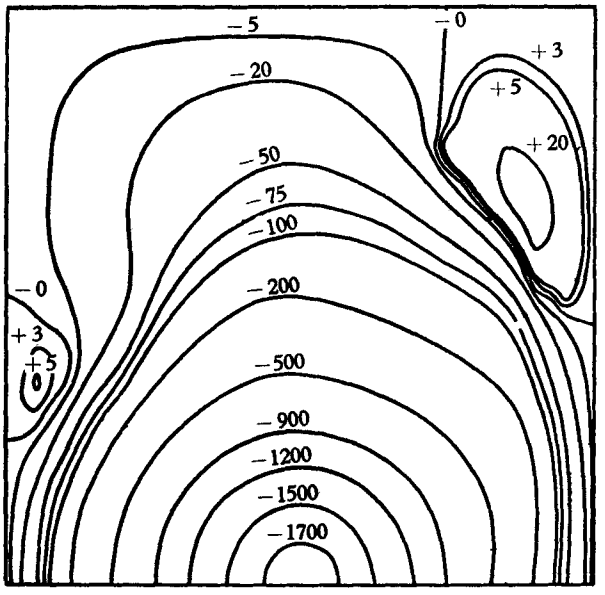

FIGURE 5. Stream-function contours for re-entrant viscous flow. Amplitude = 1000; time = 0·022.

so that it is not surprising that the flow patterns depicted in these various figures are unsymmetrical.

The author wishes to express his appreciation to Dr Robin Esch for numerous stimulating and helpful discussions during the course of the development of the present method.

REFERENCES

DIX, D. M. 1963 *J. Fluid Mech.* **15**, 449.

DUFORT, E. C. & FRANKEL, S. P. 1953 *M.T.A.C.* **7**, 135.

FORSYTHE, G. E. & WASOW, W. R. 1960 *Finite Difference Methods.* New York: Wiley.

FROMM, J. E. & HARLOW, F. H. 1963 *Phys. Fluids*, **6**, 975.

HELLUMS, J. D. & CHURCHILL, S. W. 1961 *Proc. Int. Heat Transf. Conf.* 985.

PEACEMAN, D. W. & RACHFORD, H. H. 1955 *J.S.I.A.M.* **3**, 28.

PEARSON, C. E. 1964 *A Computational Method for Viscous Flow Problems.* Sperry Rand Research Center.

PEARSON, C. E. 1965 *J. Fluid Mech.* **21**, 623.

THOMPSON, P. D. 1961 *Numerical Weather Analysis and Prediction.* New York: MacMillan.

WILKES, J. O. 1963 Thesis, Univ. of Michigan.