Numerical Solution
of the Navier-Stokes Equations*

By Alexandre Joel Chorin

Abstract. A finite-difference method for solving the time-dependent Navier-Stokes equations for an incompressible fluid is introduced. This method uses the primitive variables, i.e. the velocities and the pressure, and is equally applicable to problems in two and three space dimensions. Test problems are solved, and an application to a three-dimensional convection problem is presented.

Introduction. The equations of motion of an incompressible fluid are

$$\partial_t u_i + u_j \partial_j u_i = -\frac{1}{\rho_0} \partial_i p + \nu \nabla^2 u_i + E_i, \qquad \left(\nabla^2 \equiv \sum_j \partial_j^2 \right), \qquad \partial_j u_j = 0,$$

where u_i are the velocity components, p is the pressure, ρ_0 is the density, E_i are the components of the external forces per unit mass, ν is the coefficient of kinematic viscosity, t is the time, and the indices i, j refer to the space coordinates $x_i, x_j, i, j = 1, 2, 3$. ∂_i denotes differentiation with respect to x_i, and ∂_t differentiation with respect to the time t. The summation convention is used in writing the equations.
We write

$$u_i' = \frac{u_i}{U}, \qquad x_i' = \frac{x_i}{d}, \qquad p' = \left(\frac{d}{\rho_0 \nu U} \right) p$$

$$E_i' = \left(\frac{\nu U}{d^2} \right) E_i, \qquad t' = \left(\frac{\nu}{d^2} \right) t,$$

where U is a reference velocity, and d a reference length. We then drop the primes. The equations become

(1) $$\partial_t u_i + R u_j \partial_j u_i = -\partial_i p + \nabla^2 u_i + E_i,$$

(2) $$\partial_j u_j = 0,$$

where $R = Ud/\nu$ is the Reynolds number. It is our purpose to present a finite-difference method for solving these equations in a bounded region $\mathfrak{D}$, in either two- or three-dimensional space. The distinguishing feature of this method lies in the use of Eqs. (1) and (2), rather than higher-order derived equations. This makes it possible to solve the equations and to satisfy the imposed boundary conditions while achieving adequate computational efficiency, even in problems involving three space variables and time. The author is not aware of any other method for which such claims can be made.

Principle of the Method. Equation (1) can be written in the form

(1)′
$$\partial_t u_i + \partial_i p = \mathfrak{F}_i u ,$$

where $\mathfrak{F}_i u$ depends on u_i and E_i but not on p; Eq. (2) can be differentiated to yield

(2)′
$$\partial_i(\partial_t u_i) = 0 .$$

The proposed method can be summarized as follows: the time t is discretized; at every time step $\mathfrak{F}_i u$ is evaluated; it is then decomposed into the sum of a vector with zero divergence and a vector with zero curl. The component with zero divergence is $\partial_t u_i$, which can be used to obtain u_i at the next time level; the component with zero curl is $\partial_i p$. This decomposition exists and is uniquely determined whenever the initial value problem for the Navier-Stokes equations is well posed; it has also been extensively used in existence and uniqueness proofs for the solution of these equations (see e.g. [1]).

Let u_i, p denote not only the solution of (1) and (2) but also its discrete approximation, and let Du be a difference approximation to $\partial_j u_j$. It is assumed that at time $t = n\Delta t$ a velocity field u_i^n is given, satisfying $Du^n = 0$. The task at hand is to evaluate u_i^{n+1} from Eq. (1), so that $Du^{n+1} = 0$.

Let $Tu_i \equiv bu_i^{n+1} - Bu_i$ approximate $\partial_t u_i$, where b is a constant and Bu_i a suitable linear combination of u_i^{n-j}, $j \geqq 0$. An auxiliary field u_i^{aux} is first evaluated through

(3)
$$bu_i^{\text{aux}} - Bu_i = F_i u$$

where $F_i u$ approximates $\mathfrak{F}_i u$. u_i^{aux} differs from u_i^{n+1} because the pressure term and Eq. (2) have not been taken into account. u_i^{aux} may be evaluated by an implicit scheme, i.e. $F_i u$ may depend on u_i^n, u_i^{aux} and intermediate fields, say u_i^*, u_i^{**}. $bu_i^{\text{aux}} - Bu_i$ now approximates $\mathfrak{F}_i u$ to within an error which may depend on Δt.

Let $G_i p$ approximate $\partial_i p$. To obtain u_i^{n+1}, p^{n+1} it is necessary to perform the decomposition

$$F_i u = bu_i^{\text{aux}} - Bu_i = Tu_i + G_i p^{n+1} , \qquad D(Tu) = 0 .$$

It is, however, assumed that $Du^{n-j} = 0$, $j \geqq 0$. It is necessary therefore only to perform the decomposition

(4)
$$u_i^{\text{aux}} = u_i^{n+1} + b^{-1}G_i p^{n+1} ,$$

where $Du_i^{n+1} = 0$, and u_i^{n+1} satisfies the prescribed boundary conditions. Since p^n is usually available and is a good first guess for the values of p^{n+1}, the decomposition (4) is probably best done by iteration. For that purpose we introduce the following iteration scheme:

(5a)
$$u_i^{n+1,m+1} = u_i^{\text{aux}} - b^{-1}G_i^m p , \qquad m \geqq 1 ,$$

(5b)
$$p^{n+1,m+1} = p^{n+1,m} - \lambda Du^{n+1,m+1} , \qquad m \geqq 1 ,$$

where λ is a parameter, $u_i^{n+1,m+1}$ and $p^{n+1,m+1}$ are successive approximations to u_i^{n+1}, p^{n+1}, and $G_i^m p$ is a function of $p^{n+1,m+1}$ and $p^{n+1,m}$ which converges to $G_i p^{n+1}$ as $|p^{n+1,m+1} - p^{n+1,m}|$ tends to zero. We set

$$p^{n+1,1} = p^n .$$

The iterations (5a) are to be performed in the interior of $\mathfrak{D}$, and the iterations (5b) in $\mathfrak{D}$ and on its boundary.

It is evident that (5a) tends to (4) if the iterations converge. We are using $G_i{}^m p$ instead of $G_i p$ in (5a) so as to be able to improve the rate of convergence of the iterations. This will be discussed in detail in a later section. The form of Eq. (5b) was suggested by experience with the artificial compressibility method [2] where, for the purpose of finding steady solutions of Eqs. (1) and (2), p was related to u_i by the equation

$$\partial_t p = \text{constant} \cdot (\partial_j u_j) .$$

When for some l and a small predetermined constant ϵ

$$\max_{\mathfrak{D}} |p^{n+1,\,l+1} - p^{n+1,\,l}| \leqq \epsilon$$

we set

$$u_i{}^{n+1} = u_i{}^{n+1,\,l+1} , \qquad p^{n+1} = p^{n+1,\,l+1} .$$

The iterations (5) ensure that Eq. (1), including the pressure term, is satisfied inside $\mathfrak{D}$, and Eq. (2) is satisfied in $\mathfrak{D}$ and on its boundary.

The question of stability and convergence for methods of this type has not been fully investigated. I conjecture that the over-all scheme which yields $u_i{}^{n+1}$ in terms of $u_i{}^n$ is stable if the scheme

$$T u_i = F_i u$$

is stable. The numerical evidence lends support to this conjecture.

We shall now introduce specific schemes for evaluating $u_i{}^{\text{aux}}$ and specific representations for Du, $G_i p$, $G_i{}^m p$. Many other schemes and representations can be found. The ones we shall be using are efficient, but suitable mainly for problems in which the boundary data are smooth and the domain $\mathfrak{D}$ has a relatively simple shape.

Evaluation of u_i^{aux}. We shall first present schemes for evaluating u_i^{aux}, defined by (3).

Equation (3) represents one step in time for the solution of the Burgers equation

$$\partial_t u_i = \mathfrak{F}_i u ,$$

which can be approximated in numerous ways. We have looked for schemes which are convenient to use, implicit, and accurate to $O(\Delta t) + O(\Delta x^2)$, where Δx is one of the space increments Δx_i, $i = 1, 2, 3$. Implicit schemes were sought because explicit ones typically require, in three space dimensions, that

$$\Delta t < \tfrac{1}{6} \Delta x^2$$

which is an unduly restrictive condition. On the other hand, implicit schemes of accuracy higher than $O(\Delta t)$ would require the solution of nonlinear equations at every step, and make it necessary to evaluate $u_i{}^{\text{aux}}$ and $u_i{}^{n+1}$ simultaneously

rather than in succession. Since we assume throughout that $\Delta t = O(\Delta x^2)$, the gain in accuracy would not justify the effort.

Two schemes have been retained after some experimentation. For both of them

$$Tu_i \equiv (u_i{}^{n+1} - u_i{}^n)/\Delta t, \qquad (b^{-1} \equiv \Delta t, \quad Bu_i \equiv u_i{}^n/\Delta t).$$

They are both variants of the alternating direction implicit method.

(A) In two-dimensional problems we use a Peaceman-Rachford scheme, as proposed by Wilkes in [3] in a different context. This takes the form

$$u^*_{i(q,r)} = u^n_{i(q,r)} - R\frac{\Delta t}{4\Delta x_1} u^n_{1(q,r)}(u^*_{i(q+1,r)} - u^*_{i(q-1,r)})$$

$$- R\frac{\Delta t}{4\Delta x_2} u^n_{2(q,r)}(u^n_{i(q,r+1)} - u^n_{i(q,r-1)})$$

(6a)
$$+ \frac{\Delta t}{2\Delta x_1{}^2} (u^*_{i(q+1,r)} + u^*_{i(q-1,r)} - 2u^*_{i(q,r)})$$

$$+ \frac{\Delta t}{2\Delta x_2{}^2} (u^n_{i(q,r+1)} + u^n_{i(q,r-1)} - 2u^n_{i(q,r)})$$

$$+ \frac{\Delta t}{2} E_i,$$

$$u^{\text{aux}}_{i(q,r)} = u^*_{i(q,r)} - R\frac{\Delta t}{4\Delta x_1} u^*_{1(q,r)}(u^*_{i(q+1,r)} - u^*_{i(q-1,r)})$$

$$- R\frac{\Delta t}{4\Delta x_2} u^*_{2(q,r)}(u^{\text{aux}}_{i(q,r+1)} - u^{\text{aux}}_{i(q,r-1)})$$

(6b)
$$+ \frac{\Delta t}{2\Delta x_1{}^2} (u^*_{i(q+1,r)} + u^*_{i(q-1,r)} - u^*_{i(q,r)})$$

.
$$+ \frac{\Delta t}{2\Delta x_2{}^2} (u^{\text{aux}}_{i(q,r+1)} + u^{\text{aux}}_{i(q,r-1)} - 2u^{\text{aux}}_{i(q,r)})$$

$$+ \frac{\Delta t}{2} E_i,$$

where $u_i{}^*$ are auxiliary fields, and $u_{i(q,r)} \equiv u_i(q\Delta x_1, r\Delta x_2)$. As usual, the one-dimensional systems of algebraic equations can be solved by Gaussian elimination.

(B) In two-dimensional and three-dimensional problems we use a variant of the alternating direction method analyzed by Samarskii in [4]. This takes the form

$$u^*_{i(q,r,s)} = u^n_{i(q,r,s)} - R\frac{\Delta t}{2\Delta x_1} u^n_{1(q,r,s)}(u^*_{i(q+1,r,s)} - u^*_{i(q-1,r,s)})$$

$$+ \frac{\Delta t}{\Delta x_1{}^2} (u^*_{i(q+1,r,s)} + u^*_{i(q-1,r,s)} - 2u^*_{i(q,r,s)}),$$

$$u^{**}_{i(q,r,s)} = u^*_{i(q,r,s)} - R\frac{\Delta t}{2\Delta x_2} u^*_{2(q,r,s)}(u^{**}_{i(q,r+1,s)} - u^{**}_{i(q,r-1,s)})$$

$$+ \frac{\Delta t}{\Delta x_2{}^2} (u^{**}_{i(q,r+1,s)} + u^{**}_{i(q,r-1,s)} - 2u^{**}_{i(q,r,s)})$$

$$u_{i(q,r,s)}^{\text{aux}} = u_{i(q,r,s)}^{**} - R\frac{\Delta t}{2\Delta x_3}u_{3(q,r,s)}^{**}(u_{i(q,r,s+1)}^{\text{aux}} - u_{i(q,r,s-1)}^{\text{aux}})$$

$$+ \frac{\Delta t}{\Delta x_3^2}(u_{i(q,r,s+1)}^{\text{aux}} + u_{i(q,r,s-1)}^{\text{aux}} - 2u_{i(q,r,s)}^{\text{aux}})$$

$$+ \Delta t E_{i(q,r,s)},$$

$$u_{i(q,r,s)} \equiv u_i(q\Delta x_1, r\Delta x_2, s\Delta x_3),$$

$$E_{i(q,r,s)} \equiv E_i(q\Delta x_1, r\Delta x_2, s\Delta x_3).$$

u_i^*, u_i^{**} are auxiliary fields. These equations can be written in the symbolic form

(7)
$$(I - \Delta t Q_1)u_i^* = u_i^n,$$
$$(I - \Delta t Q_2)u_i^{**} = u_i^*,$$
$$(I - \Delta t Q_3)u_i^{\text{aux}} = u_i^{**} + \Delta t E_i,$$

where I is the identity operator, and Q_l involves differentiations with respect to the variable x_l only.

It can be verified that when $R = 0$ scheme (6) is accurate to $O(\Delta t^2) + O(\Delta x^2)$. When $R \neq 0$ however, they are both accurate to the same order. Scheme (7) is stable in three-dimensional problems; the author does not know of a simple extension of scheme (6) to the three-dimensional case. Scheme (7) has two useful properties: It requires fewer arithmetic operations per time step than scheme (6), and because of the simple structure of the right-hand sides, the intermediate fields u_i^*, u_i^{**} do not have to be stored separately.

If either scheme is to be used in a problem in which the velocities u_i^{n+1} are prescribed at the boundary, values of u_i^*, u_i^{**}, u_i^{aux} at the boundary have to be provided in advance so that the several implicit operators can be inverted. Consider the case of the scheme (7). We have

$$u_i^{n+1} = (I + \Delta t Q_1 + \Delta t Q_2 + \Delta t Q_3)u_i^n + \Delta t E_i - \Delta t G_i p^n + O(\Delta t^2),$$
$$u_i^* = (I + \Delta t Q_1)u_i^n + O(\Delta t^2),$$
$$u_i^{**} = (I + \Delta t Q_1 + \Delta t Q_2)u_i^n + O(\Delta t^2),$$
$$u_i^{\text{aux}} = (I + \Delta t Q_1 + \Delta t Q_2 + \Delta t Q_3)u_i^n + \Delta t E_i + O(\Delta t^2).$$

From these relations it can be deduced that if we set at the boundary

(8)
$$u_i^* = u_i^{n+1} - \Delta t Q_2 u_i^{n+1} - \Delta t Q_3 u_i^{n+1} - \Delta t E_i + \Delta t \overline{G}_i p,$$
$$u_i^{**} = u_i^{n+1} - \Delta t Q_3 u_i^{n+1} - \Delta t E_i + \Delta t \overline{G}_i p,$$
$$u_i^{\text{aux}} = u_i^{n+1} + \Delta t G_i p,$$

the scheme will remain accurate to $O(\Delta t)$. Here $\overline{G}_i$ does not have to be identical with G_i; all we need is

$$\overline{G}_i p^n = G_i p^n + O(\Delta t).$$

The reason for introducing the new operator $\overline{G}_i$ is that at the boundary the normal component of G_i has to be approximated by one-sided differences, while this is not necessary in the interior of the domain $\mathfrak{D}$ where Eq. (4) is assumed to hold.

More accurate expressions for the auxiliary fields at the boundaries can be

used, provided one is willing to invest the additional programming effort required to implement them on the computer. Appropriate expressions for $u_i{}^*$, $u_i{}^{aux}$ at the boundary can be derived for use with the scheme (6).

It should be noted that for problems in which the viscosity is negligible, it is possible to devise explicit schemes accurate to $O(\Delta t^2) + O(\Delta x^2)$ and stable when $\Delta t = O(\Delta x)$. Such schemes will be discussed elsewhere.

The Dufort-Frankel Scheme and Successive Point Over-Relaxation. In order to explain our construction of D, $G_i{}^m$ and our choice of λ for use in (5a) and (5b), we need a few facts concerning the Dufort-Frankel scheme for the heat equation and its relation to the relaxation method for solving the Laplace equation.

Consider the equation

$$(9) \qquad\qquad -\nabla^2 u = f, \qquad (\nabla^2 \equiv \partial_1^2 + \partial_1^2)$$

in some nice domain $\mathfrak{D}$, say a rectangle. u is assumed known on the boundary of $\mathfrak{D}$. We approximate this equation by

$$(10) \qquad\qquad -Lu = f,$$

where L is the usual five-point approximation to the Laplacian, and u and f are now m-component vectors. m is the number of internal nodes of the resulting difference scheme. For the sake of simplicity we assume that the mesh spacings in the x_1 and x_2 directions are equal, $\Delta x_1 = \Delta x_2 = \Delta x$; this implies no essential restriction. The operator $-L$ is represented by an $m \times m$ matrix A.

We write

$$A = A' - E - E'$$

where E, E' are respectively strictly upper and lower triangular matrices, and A' is diagonal. The convergent relaxation iteration scheme for solving (10) is defined by

$$(11) \qquad (A' - \omega E)u^{n+1} = \{(1 - \omega)A' + \omega E'\}u^n + \omega f$$

(see e.g. Varga [5]). ω is the relaxation factor, $0 < \omega < 2$, and the u^n are the successive iterates. The evaluation of the optimal relaxation factor ω_{opt} depends on the fact that A satisfies "Young's condition (A)," i.e. that there exists a permutation matrix P such that

$$(12) \qquad\qquad P^{-1}AP = \Lambda - N,$$

where Λ is diagonal, and N has the normal form

$$\begin{pmatrix} 0 & G \\ G' & 0 \end{pmatrix}$$

the zero submatrices being square. Under this condition, ω_{opt} can be readily determined.

The matrix A depends on the order in which the components of u^{n+1} are computed from u^n. Changing that order is equivalent to transforming A into $P^{-1}AP$, where P is a permutation matrix.

We now consider the solution of (14) to be the asymptotic steady solution of

(13) $$\partial_\tau u = \nabla^2 u + f$$

and approximate the latter equation by the Dufort-Frankel scheme

$$u_{q,r}^{n+1} - u_{q,r}^{n-1} = \frac{2\Delta\tau}{\Delta x^2}\left(u_{q+1,r}^n + u_{q-1,r}^n + u_{q,r+1}^n + u_{q,r-1}^n - 2u_{q,r}^{n+1} - 2_{q,r}^{n-1}\right) + 2\Delta\tau f,$$

$$u_{q,r}^n \equiv u(q\Delta x_1, r\Delta x_2, n\Delta\tau)$$

which approximates (13) when $\Delta t = o(\Delta x)$. Grouping terms, we obtain

(14)
$$\left(1 + 4\frac{\Delta\tau}{\Delta x^2}\right)u_{q,r}^{n+1} - \left(1 - 4\frac{\Delta\tau}{\Delta x^2}\right)u_{q,r}^{n-1}$$
$$= 2\frac{\Delta\tau}{\Delta x^2}\left(u_{q+1,r}^n + u_{q-1,r}^n + u_{q,r+1}^n + u_{q,r-1}^n\right) + 2\Delta\tau f.$$

Since $u_{q,r}^n$ does not appear in (14), the calculation separates into two independent calculations on intertwined meshes, one of which can be omitted. When this is done, we can write

$$U^{n+1} = \begin{pmatrix} u^{2n} \\ u^{2n+1} \end{pmatrix} \qquad (U^{n+1} \text{ has } m \text{ components}) .$$

If we then write

(15) $$\omega = \frac{8\Delta\tau/\Delta x^2}{1 + 4\Delta\tau/\Delta x^2}$$

we see that the iteration (14) reduces to an iteration of the form (11) where the new components of U^{n+1} are calculated in an order such that A has the normal form (12). The Dufort-Frankel scheme appears therefore to be a particular ordering of the over-relaxation method whose existence is equivalent to Young's condition (A).

The best value of $\Delta\tau$, $\Delta\tau_{\text{opt}}$, can be determined from ω_{opt} and relation (15). We find that $\Delta\tau_{\text{opt}} = O(\Delta x)$, therefore for $\Delta\tau = \Delta\tau_{\text{opt}}$ the Dufort-Frankel scheme approximates, not Eq. (13), but rather the equation

$$\partial_\tau u = \nabla^2 u - 2\left(\frac{\Delta\tau}{\Delta x}\right)^2 \partial_\tau^2 u + f .$$

This is the equation which Garabedian in [6] used to estimate ω_{opt}. It can be used here to estimate $\Delta\tau_{\text{opt}}$. These remarks obviously generalize to problems where $\Delta x_1 \neq \Delta x_2$ or where there are more than two space variables.

The following remark will be of use: We could have approximated Eq. (13) by the usual explicit formula

(16) $$u_{q,r}^{n+1} - u_{q,r}^n = \frac{\Delta\tau}{\Delta x^2}\left(u_{q+1,r}^n + u_{q-1,r}^n + u_{q,r+1}^n + u_{q,r-1}^n - 4u_{q,r}^n\right) + \Delta\tau f$$

and used this formula as an iteration procedure for solving (10). The resulting iteration converges only when $\Delta\tau/\Delta x^2 < 1/4$, and the convergence is very slow. The rapidly converging iteration procedure (14) can be obtained from (16) by splitting the term $u_{q,r}^n$ on the right-hand side into $\frac{1}{2}(u_{q,r}^{n+1} + u_{q,r}^{n-1})$.

Representation of D, G_i and G_i^m, and the Iteration Procedure for Determining
u_i^{n+1}, p^{n+1}. For the sake of clarity we shall assume in this section that the domain
$\mathfrak{D}$ is two-dimensional and rectangular, and that the velocities are prescribed at the
boundary. Extension of the procedure to three-dimensional problems is immedi-
ate, and extension to problems with other types of boundary conditions often pos-
sible. Stress-free boundaries and periodicity conditions in particular offer no dif-
ficulty. Domains of more complicated shape can be treated with the help of ap-
propriate interpolation procedures.

Our first task is to define D. Let $\mathfrak{B}$ denote the boundary of $\mathfrak{D}$ and $\mathfrak{C}$ the set of
mesh nodes with a neighbor in $\mathfrak{B}$. In $\mathfrak{D} - \mathfrak{B}$ we approximate the equation of con-
tinuity by centered differences, i.e. we set

$$(17) \qquad Du \equiv \frac{1}{2\Delta x_1}\left(u_{1(q+1,r)} - u_{1(q-1,r)}\right) + \frac{1}{2\Delta x_2}\left(u_{2(q,r+1)} - u_{2(q,r-1)}\right) = 0 .$$

At the points of $\mathfrak{B}$ we use second-order one-sided differences, so that Du is
accurate to $O(\Delta x^2)$ everywhere. Consider the boundary line $x_2 = 0$, represented
by $j = 1$ (Fig. 1). We have on that line

$$(18) \qquad \begin{aligned} Du &\equiv \frac{2}{\Delta x_2}\left[u_{2(q,2)} - u_{2(q,1)} - \tfrac{1}{4}(u_{2(q,3)} - u_{2(q,1)})\right] \\ &\quad + \frac{1}{2\Delta x_1}\left(u_{1(q+1,1)} - u_{1(q-1,1)}\right) = 0 \end{aligned}$$

with similar expressions at the other boundaries. Equation (17) states that the
total flow of fluid into a rectangle of sides $2\Delta x_1$, $2\Delta x_2$ is zero. Equation (18) does
not have this elementary interpretation.
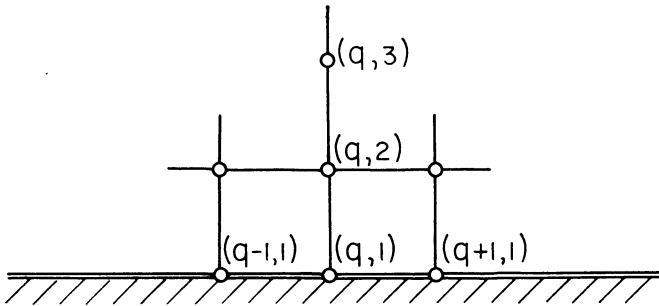
FIGURE 1. *Mesh Near a Boundary.*

We now define $G_i p$ at every point of $\mathfrak{D} - \mathfrak{B}$ by

$$G_1 p = \frac{1}{2\Delta x_1}\left(p_{q+1,r} - p_{q-1,r}\right) ,$$

$$G_2 p = \frac{1}{2\Delta x_2}\left(p_{q,r+1} - p_{q,r-1}\right) ,$$

$$p_{q,r} \equiv p(q\Delta x_1, r\Delta x_2) ,$$

i.e. $\partial_i p$ is approximated by centered differences. It should be emphasized that
these forms of $G_i p$ and Du are not the only possible ones.

It is our purpose now to perform the decomposition (4). u_i^{n+1} is given on the boundary $\mathfrak{B}$, u_i^{aux} is given in $\mathfrak{D} - \mathfrak{B}$ (the values of u_i^{aux} on $\mathfrak{B}$, used in (6) or (7), are of no further use). p^{n+1} is to be found in $\mathfrak{D}$ (including the boundary) and u^{n+1} in $\mathfrak{D} - \mathfrak{B}$, so that in $\mathfrak{D} - \mathfrak{B}$

$$u_i^{\text{aux}} = u_i^{n+1} + \Delta t G_i p$$

and in $\mathfrak{D}$ (including the boundary)

$$Du^{n+1} = 0 \, .$$

This is to be done using the iterations (5), where the form of $G_i^m p$ has not yet been specified.

At a point (q, r) in $\mathfrak{D} - \mathfrak{B} - \mathfrak{C}$, i.e. far from the boundary, one can substitute Eq. (5a) into Eq. (5b), and obtain

(19) $$p^{n+1, m+1} - p^{n+1, m} = -\lambda Du^{\text{aux}} + \Delta t \lambda DG^m p \, .$$

This is an iterative procedure for solving the equation

(20) $$Lp = \frac{1}{\Delta t} Du^{\text{aux}} \, ,$$

where $Lp \equiv DGp$ approximates the Laplacian of p. With our choice of D and G_i, Lp is a five-point formula using a stencil whose nodes are separated by $2\Delta x_1$, $2\Delta x_2$. Equation (20) is of course a finite-difference analogue of the equation

(21) $$\nabla^2 p = \partial_i \partial_j u_i u_j + \partial_j E_j \, ,$$

which can be obtained from Eq. (1) by taking its divergence. At points of $\mathfrak{B}$ or $\mathfrak{C}$ if it is not possible to substitute (5a) into (5b) because at the boundary u_i^{n+1} is prescribed, $u^{n+1, m+1} = u^{n+1}$ for all m, (5a) does not hold and therefore (19) is not true. Near the boundary the iterations (5) provide boundary data for (20) and ensure that the constraint of incompressibility is satisfied. We proceed as follows: $G_i^m p$ and λ are chosen so that (19) is a rapidly converging iteration for solving (20); $G_i^m p$ at the boundary are then chosen so that the iterations (5) converge everywhere.

Let (q, r) again be a node in $\mathfrak{D} - \mathfrak{B} - \mathfrak{C}$. $u_i^{n+1, m}$ and $p^{n+1, m}$ are assumed known. We shall evaluate simultaneously $p_{q,r}^{n+1, m+1}$ and the velocity components involved in the equation $Du^{n+1} = 0$ at (q, r), i.e. $u_{1(q\pm1, r)}^{n+1, m+1}$, $u_{2(q, r\pm1)}^{n+1, m+1}$ (Fig. 2). These velocity components depend on the value of p at (q, r) and on the values of p at other points. Following the spirit of the remark at the end of the last section, the value of p at (q, r) is taken to be

$$\tfrac{1}{2}(p_{q,r}^{n+1, m+1} + p_{q,r}^{n+1, m})$$

while at other points we use $p^{n+1, m}$.

This leads to the following formulae

(22a) $$p_{q,r}^{n+1, m+1} = p_{q,r}^{n+1, m} - \lambda Du^{n+1, m+1}$$

$$(D \text{ given by } (17))$$

(22b) $$u_{1(q+1, r)}^{n+1, m+1} = u_{1(q+1, r)}^{\text{aux}} - \frac{\Delta t}{2\Delta x_1} (p_{q+2, r}^{n+1, m} - \tfrac{1}{2}(p_{q,r}^{n+1, m+1} + p_{q,r}^{n+1, m})) \, ,$$

(22c) $$u_{1(q-1,r)}^{n+1,m+1} = u_{1(q-1,r)}^{\text{aux}} - \frac{\Delta t}{2\Delta x_1} (\tfrac{1}{2}(p_{q,r}^{n+1,m+1} + p_{q,r}^{n+1,m}) - p_{q-2,r}^{n+1,m}) ,$$

(22d) $$u_{2(q,r+1)}^{n+1,m+1} = u_{2(q,r+1)}^{\text{aux}} - \frac{\Delta t}{2\Delta x_2} (p_{q,r+2}^{n+1,m} - \tfrac{1}{2}(p_{q,r}^{n+1,m+1} + p_{q,r}^{n+1,m})) ,$$

(22e) $$u_{2(q,r-1)}^{n+1,m+1} = u_{2(q,r-1)}^{\text{aux}} - \frac{\Delta t}{2\Delta x_2} (\tfrac{1}{2}(p_{q,r}^{n+1,m+1} + p_{q,r}^{n+1,m}) - p_{q,r-2}^{n+1,m}) .$$

These equations define $G_i{}^m p$. Clearly, $G_i{}^m p \rightarrow G_i p$.

Equations (22) can be solved for $p_{q,r}^{n+1,m+1}$, yielding

(23a) $$p_{q,r}^{n+1,m+1} = (1 + \alpha_1 + \alpha_2)^{-1}[(1 - \alpha_1 - \alpha_2)p^{n+1,m} - \lambda D u^{\text{aux}}$$
$$+ \alpha_1(p_{q+2,r}^{n+1,m} + p_{q-2,r}^{n+1,m}) + \alpha_2(p_{q,r+1}^{n+1,m} + p_{q,r-2}^{n+1,m})] ,$$

where $\alpha_i = \lambda \Delta t/4 \Delta x_i$, $i = 1, 2$. This can be seen to be a Dufort-Frankel relaxation scheme for the solution of (20), as was to be expected. The $\Delta \tau$ of the preceding section is replaced here by $\lambda \Delta t/2$. Corresponding to $\Delta \tau_{\text{opt}}$ (or ω_{opt}) we find λ_{opt}. If p were known on $\mathcal{B}$ and $\mathcal{C}$, convergence of the iterations (23a) would follow from the discussion in the preceding section, and $\lambda = \lambda_{\text{opt}}$ would lead to the highest rate of convergence.
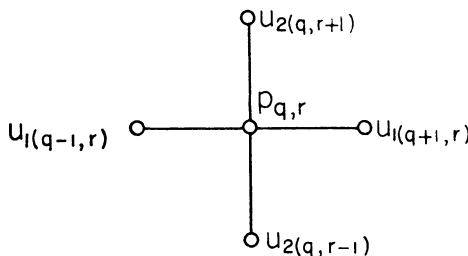
FIGURE 2. *Iteration Scheme*

In $\mathcal{B}$ and $\mathcal{C}$ formulae (22) are modified by the use of the known values of u_i^{n+1} at the boundary whenever necessary. This leads to the formulae, for (q, r) in $\mathcal{C}$:

(23b) $$p_{q,2}^{n+1,m+1} = (1 + \alpha_1 + \tfrac{1}{2}\alpha_2)^{-1}[(1 - \alpha_1 - \tfrac{1}{2}\alpha_2)p_{q,2}^{n+1,m} - \lambda D u^{\text{aux}}$$
$$+ \alpha_1(p_{q+2,2}^{n+1,m} + p_{q-2,2}^{n+1,m}) + \alpha_2 p_{q,4}^{n+1,m}] ,$$

and for (q, r) on $\mathcal{B}$:

(23c) $$p_{q,1}^{n+1,m+1} = (1 + \alpha_1)^{-1}[(1 - \alpha_1)p_{q,1}^{n+1,m} - \lambda D u^{\text{aux}}$$
$$+ 2\alpha_2(p_{q,3}^{n+1,m} - \tfrac{1}{4}(p_{q,4}^{n+1,m} - p_{q,2}^{n+1,m}))]$$

etc. In (23b) Du is given by (17), and in (23c) by (18). u_i^{aux} at the boundary is interpreted as u_i^{n+1}. Although no proof is offered, a heuristic argument and the numerical evidence lead us to state that the whole iteration system—Eqs. (23a), (23b), (23c)—converges for all $\lambda > 0$ and converges fastest when $\lambda \sim \lambda_{\text{opt}}$. None of the boundary instabilities which arise in two-dimensional vorticity-stream function calculation has been observed.

It can be seen that because our representation of $Du = 0$ expresses the balance of mass in a rectangle of sides $2\Delta x_i$, $i = 1, 2$, the pressure iterations split into

two calculations on intertwined meshes, coupled at the boundary. The most efficient orderings for performing the iterations are such that the resulting over-all scheme is a Dufort-Frankel scheme for each one of the intertwined meshes. This involves no particular difficulty; a possible ordering for a rectangular grid is shown in Fig. 3. The iterations are to be performed until for some l

$$\max_{q,r} |p_{q,r}^{n+1,\,l+1} - p_{q,r}^{n+1,\,l}| \leqq \epsilon$$

for a predetermined ϵ.

The new velocities u_i^{n+1}, $i = 1, 2$, are to be evaluated using (22b), (22c), (22d), (22e). This has to be done only after the $p^{n+1,m}$ have converged. There is no need to evaluate and store the intermediate fields $u_i^{n+1,m+1}$. A saving in computing time can be made by evaluating Du^{aux} at the beginning of each iteration. We notice two advantages of our iteration procedure: Du^{n+1} can be made as small as one wishes independently of the error in Du^n; and when $p^{n+1,l+1}$ and $p^{n+1,l}$ differ by less than ϵ, $Du^{n+1} = O(\epsilon/\lambda)$; it can be seen that $\lambda_{opt} = O(\Delta x^{-1})$, hence $Du^{n+1} = O(\epsilon\Delta x)$. A gain in accuracy appears, which can be used to relax the convergence criterion for the iterations. This gain in accuracy is due to the fact that the u_i^{n+1} are evaluated using an appropriate combination of $p^{n+1,l}$ and $p^{n+1,l+1}$, rather than only the latest iterate $p^{n+1,l+1}$.
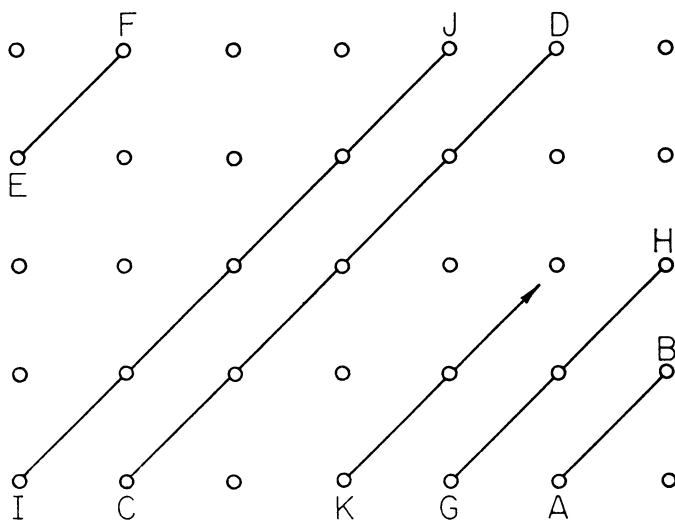
The domain is swept in the order AB, CD, EF, GH, IJ, K

FIGURE 3. *An Ordering for the Iteration Scheme*

Solution of a Simple Test Problem. The proposed method was first applied to a simple two-dimensional test problem, used as a test problem by Pearson in [7] for a vorticity-stream function method. $\mathfrak{D}$ is the square $0 \leqq x_i \leqq \pi$, $i = 1, 2$; $E_1 = E_2 = 0$; the boundary data are

$$u_1 = -\cos x_1 \sin x_2 e^{-2t}, \qquad u_2 = \sin x_1 \cos x_2 e^{-2t}.$$

The initial data are

$$u_1 = -\cos x_1 \sin x_2 , \qquad u_2 = \sin x_1 \cos x_2 .$$

The exact solution of the problem is

$$u_1 = -\cos x_1 \sin x_2 e^{-2t} , \qquad u_2 = \sin x_1 \cos x_2 e^{-2t} ,$$
$$p = -R \tfrac{1}{4}(\cos 2x_1 + \cos 2x_2)e^{-4t} ,$$

where R is the Reynolds number. This solution has the property

$$\partial_i p = -R u_j \partial_j u_i ;$$

hence curl (u_i) satisfies a linear equation. Nevertheless, this problem is a fair test of our method because $Du^{\text{aux}} \neq 0$.

We first evaluate λ_{opt}. For the equation

$$-Lu = f$$

in $\mathfrak{D}$, with a grid of mesh widths $2\Delta x_1$, $2\Delta x_2$, and u known on the boundary, we have

$$\omega_{\text{opt}} = \frac{2}{1 + (1 - \alpha^2)^{1/2}} ,$$

where $\alpha = \tfrac{1}{2}(\cos 2\Delta x_1 + \cos 2\Delta x_2)$ is the largest eigenvalue of the associated Jacobi matrix (see [5]).

We put

$$q = \frac{\lambda_{\text{opt}}}{2}\left(\frac{\Delta t}{\Delta x_1^2} + \frac{\Delta t}{\Delta x_2^2}\right).$$

Equation (15) can be written as

$$\omega_{\text{opt}} = \frac{8q}{1 + 4q} ,$$

therefore

$$q = \frac{1}{(1 - \alpha^2)^{1/2}}$$

and

$$\lambda_{\text{opt}} = \frac{4}{(\Delta t/\Delta x_1^2 + \Delta t/\Delta x_2^2)} \frac{1}{(1 - \alpha^2)^{1/2}} .$$

We now assume $\Delta x_1 = \Delta x_2 = \Delta x$, obtaining

$$\lambda_{\text{opt}} = \frac{2\Delta x^2}{\Delta t \sin (2\Delta x)} .$$

In Tables I, II, and III we display results of some sample calculations. n is the number of time steps; $e(u_i)$, $i = 1, 2$, are the maxima over $\mathfrak{D}$ of the differences between the exact and the computed solutions u_i. It is not clear how the error in the pressure is to be represented; p^n is defined at a time intermediate between $(n - 1)\Delta t$ and $n\Delta t$; it is proportional to R in our nondimensionalization. There

are errors in p due to the fact, discussed at the end of the preceding section, that the iterations can be stopped before the $p^{n,m}$ have truly converged. $e(p)$ in the tables represents the maximum over the grid of the differences between the exact pressure at time $n\,\Delta t$ and the computed p^n, divided by R; it is given mainly for the sake of completeness. The accuracy of the scheme is to be judged by the smallness of $e(u_i)$. l is the number of iterations; it is to be noted that the first iteration always has to be performed in order that Eq. (1) be satisfied. "Scheme A" means that $u_i{}^{aux}$ was evaluated using Eq. (6), and "Scheme B" means Eq. (7) were used.

TABLE I
Scheme A; $\Delta x = \pi/39$; $\Delta t = 2\,\Delta x^2 = 0.01397$; $\epsilon = \Delta x^2$; $R = 1$

n	$e(u_1)$	$e(u_2)$	$e(p)$	l
1	2.8×10^{-4}	2.6×10^{-4}	0.0243	1
2	2.7×10^{-4}	2.0×10^{-4}	0.0136	7
3	1.5×10^{-4}	1.3×10^{-4}	0.0069	4
4	1.8×10^{-4}	1.9×10^{-4}	0.0145	4
5	1.3×10^{-4}	1.7×10^{-4}	0.0089	5
6	1.3×10^{-4}	1.8×10^{-4}	0.0116	4
7	1.6×10^{-4}	1.9×10^{-4}	0.0144	4
9	1.4×10^{-4}	1.7×10^{-4}	0.0147	4
10	1.3×10^{-4}	1.6×10^{-4}	0.0156	4
20	1.8×10^{-4}	2.3×10^{-4}	0.0241	4

TABLE II
Scheme A; $\Delta x = \pi/39$; $\Delta t = 2\,\Delta x^2 = 0.01397$; $\epsilon = \Delta x^3$; $R = 1$

n	$e(u_1)$	$e(u_2)$	$e(p)$	l
1	8.5×10^{-5}	3.8×10^{-5}	0.0059	10
2	1.0×10^{-4}	5.7×10^{-5}	0.0067	10
3	1.0×10^{-4}	7.0×10^{-5}	0.0068	10
4	1.0×10^{-4}	7.8×10^{-5}	0.0068	10
5	1.0×10^{-4}	8.3×10^{-5}	0.0069	10
6	9.7×10^{-5}	8.6×10^{-5}	0.0070	10
7	9.4×10^{-5}	8.7×10^{-5}	0.0071	10
8	9.0×10^{-5}	8.7×10^{-5}	0.0073	10
9	8.7×10^{-5}	8.7×10^{-5}	0.0077	10
10	8.3×10^{-5}	8.5×10^{-5}	0.0082	10
20	1.0×10^{-4}	1.0×10^{-4}	0.0216	9

TABLE III
Scheme B; $\Delta x = \pi/39$; $\Delta t = \frac{1}{2}\Delta x^2 = 0.00324$; $\epsilon = \Delta x^2$; $R = 20$

n	$e(u_1)$	$e(u_2)$	$e(p)$	l
1	1.1×10^{-3}	1.2×10^{-3}	0.0217	15
3	1.9×10^{-3}	2.1×10^{-3}	0.0234	9
5	2.5×10^{-3}	2.8×10^{-3}	0.0242	9
7	3.3×10^{-3}	3.2×10^{-3}	0.0249	9
9	4.0×10^{-3}	3.5×10^{-3}	0.0253	8
20	5.8×10^{-3}	3.9×10^{-3}	0.0258	8

Tables I and II describe computations which differ only in the value of ϵ. They show that $\epsilon = \Delta x^2$ is an adequate convergence criterion. Table III indicates that fair results can be obtained even when $R\,\Delta t$ is fairly large; when $R = 20$, $\Delta x = \pi/39$, $\Delta t = \frac{1}{2}\Delta x^2$, we have

$$R \simeq 1.5\ \Delta x^{-1}\,.$$

The errors are of the order of 1%. Additional computational results were presented in [8].

Application to Thermal Convection. Suppose a plane layer of fluid, in the field of gravity, of thickness d and infinite lateral extent, is heated from below. The lower boundary $x_3 = 0$ is maintained at a temperature T_0, the upper boundary $x_3 = d$ at a temperature $T_1 < T_0$. The warmer fluid at the bottom expands and tends to move upward; this motion is inhibited by the viscous stresses.

In the Boussinesq approximation (see e.g. [9]) the equations describing the possible motions are

$$\partial_t u_i + u_j \partial_j u_i = -\frac{1}{\rho_0}\partial_i p + \nu \nabla^2 u_i - g(1 - (T - T_0))\delta_i\,,$$

$$\partial_t T + u_j \partial_j T = k\nabla^2 T\,, \qquad \partial_j u_j = 0\,,$$

where T is the temperature, k the coefficient of thermal conductivity, α the coefficient of thermal expansion, and δ_i the components of the unit vector pointing upwards.

We write

$$u_i' = \left(\frac{d}{\nu}\right)u_i\,, \qquad T'' = \frac{T - T_1}{T_0 - T_1}\,, \qquad t' = \left(\frac{\nu^2}{d}\right)t\,,$$

$$x_i' = \frac{x_i}{d}\,, \qquad p' = \frac{1}{\rho_0}\left(\frac{d}{\nu}\right)^2 p + \frac{(T_1 - T_0)dg x_3}{\nu^2}$$

and drop the primes. The equations now are

$$\partial_t u_i + u_j \partial_j u_i = -\partial_i p + \nabla^2 u_i + \frac{R^*}{\sigma}(T - 1)\delta_i\,,$$

$$\partial_t T + u_j \partial_j T = \frac{1}{\sigma}\nabla^2 T\,, \qquad \partial_j u_j = 0\,,$$

where $R^* = \alpha g d^3 (T_0 - T_1)k\nu$ is the Rayleigh number, and $\sigma = \nu/k$ the Prandtl number. The rigid boundaries are now situated at $x_3 = 0$ and $x_3 = 1$, where it is assumed that $u_i = 0$, $i = 1, 2, 3$.

It is known that for $R^* < R_c^*$, the state of rest is stable and no steady convection can arise, where $R_c^* = 1707.762$.

When $R^* = R_c^*$, steady infinitesimal convection can first appear, and the field quantities are given by

$$u_3 = CW(x_3)\phi\,,$$

$$u_i = \frac{C}{a^2}(\partial_3 W(x_3))\partial_i \phi\,, \qquad i = 1, 2\,,$$

$$T = CT(x_3)\phi$$

where $\phi = \phi(x_1, x_2)$ determines the horizontal planform of the motion and satisfies

$$(\partial_1{}^2 + \partial_2{}^2 + a^2)\phi = 0 ,$$

$W(x_3)$, $T(x_3)$ are fully determined functions of x_3, $a = 3.117$, and C is a small but undetermined amplitude.

In two-dimensional motion $u_1 = 0$ and the motion does not depend on x_1. We then have

$$\phi = \cos ax_2 .$$

The motion is periodic in x_2 with period $2\pi/a$.

The Nusselt number Nu is defined as the ratio of the total heat transfer to the heat transfer which would have occurred if no convection were present. For $R^* \leqq R_c{}^*$, $Nu = 1$. In our dimensionless variables

$$Nu = \frac{a}{2\pi} \int_0^{2\pi/a} (u_3 T - \partial_3 T)dx_2 .$$

A similar expression holds in the three-dimensional case. When the convection is steady Nu does not depend on x_3.

When $R^* > R_c{}^*$ steady cellular convection sets in. It is of interest to determine its magnitude and its spatial configuration. The problem of its magnitude, and in particular the dependence of Nu on R^* and σ, when the motion is steady, has been studied by the author in previous work [2], [10]. As to the shape of the convection cells, it is known that flows may exist in which the cells, when viewed from above, look like hexagons, or like rectangles with various ratios of length to width, or like rolls, i.e. two-dimensional convection cells (see [11]). However, only cellular structures which are stable with respect to small perturbations can persist in nature or be exhibited by our method. It has been shown, numerically by the author [10], experimentally by Koschmieder [12] and Rossby [13], theoretically, in the case of infinite σ and small perturbations, by Busse [14], that for $R^*/R_c{}^* < 10$ the preferred cellular mode is a roll. Busse showed that the rolls are stable for wave numbers in a certain range. We shall now demonstrate numerically the impermanence of hexagonal convection and the emergence of a roll.

Consider the case $R^*/R_c{}^* = 2$, $\sigma = 1$. We assume the motion to be periodic in the x_1 and x_2 directions, with periods respectively $4\pi/a\sqrt{3}$ and $4\pi/a$ (the first period is apparently in the range of stable periods for rolls as predicted by Busse). These are the periods of the hexagonal cells which could arise when $R^* = R_c{}^*$. The state of rest is perturbed by adding to the temperature in the plane $x_3 = \Delta x_3$ a multiple of the function $\phi(x_1, x_2)$ which corresponds to a hexagonal cell, and adding a small constant to the temperature on the line $x_1 = (3/4)(4\pi/a\sqrt{3})$, $x_2 = (3/4)(4\pi/a)$. We then follow the evolution of the convection in time, using a net of $24 \times 24 \times 25$, i.e.

$$\Delta x_1 = (4\pi/a\sqrt{3})/24 , \qquad \Delta x_2 = (4\pi/a)/24 , \qquad \Delta x_3 = 1/24 .$$

We choose $\epsilon = \Delta x_2{}^2$, $\Delta t = 3\Delta x_3{}^2$. The convection pattern is visualized as follows: the velocities in the plane $x_3 = 17\Delta x_3$ are examined. If $u_{3(q,r,18)} > 0$ an $*$ is printed, if $u_{3(q,r,18)} \leqq 0$, a 0 is printed.

ALEXANDRE JOEL CHORIN

FIGURE 4. *Evolution of a Convection Cell*

```
*******000000000000000**          *********0000000000000**
*******00000000000000000          *********000000000*****
*******0000000000000****          **********0000000*******
******00000000000000*0**          ********000000000*******
****00000000000000000***          ******000000000000******
000000000000000000000000          ***000000000000000000***
000000000000000000000000          000000000000000000000000
000000000000000000000000          000000000******00000000
0000000000*0*00000000000          00000**********0000000
000000000000000000000000          0000**********000000000
000000*****0*00000000000          000*********00000000000
000000000000000000000000          000********0000000000000
000000000000000000000000          0000*****000000000000000
000000000000000000000000          000000000000000000000000
000000000000000000000000          000000000000000000000000
000000000000000000000000          00000000000000*******000
00000000000000000***00000          0000000000000********00
00000000000000000***00000          0000000000000********00
00000000000000000***00000          0000000000000********00
000000000000000000000000          00000000000000*******000
000000000000000000000000          000000000000000000000000
000000000000000000000000          00000*00000000000000000
000000000000000000000000          00****00000000000000000
000000000000000000000000          *******0000000000000000
```

4a. *After* 1 *step* $(Nu = 1)$ 4b. *After* 10 *steps* $(Nu = 1)$

```
*********00000000000000          *********00000000000000
**********000000000000*          **********000000000000*
**********000000000000*          ***********0000000000**
**********000000000****          ************000000000**
***********************          ************00000000**
***0000000*************          *************00000000**
***********************          *************00000000**
************0000000***          *************0000000000*
**********000000000000          ***********00000000000
**********0000000000000          00*********000000000000
0*********0000000000000          0000*******000000000000
00********00000000000000          0000000****000000000000
00000***000000000000000          00000000*****000000000000
000000000000******0000          00000000******000000000
00000000000**********00          00000000*********000000
00000000000**********0          00000000***********000
0000000000*************          00000000***********00
0000000000*************          00000000*************0
0000000000*************          00000000*************0
0000000000*************          00000000*************0
0000000000*************          00000000*************0
000000000000*********00          0000000000**********00
000000000000000000000000          000000000000000000000000
00******0000000000000000          000000000000000000000000
0*********00000000000000          00*****0000000000000000
```

4c. *After* 125 *steps* $(Nu = 1.25)$ 4d. *After* 225 *steps* $(Nu = 1.72)$

```
0000********000000000000          0000**********000000000
00**********0000000000          0000**********000000000
0************000000000          0000**********000000000
0************00000000          0000**********00000000
0************00000000          000***********00000000
0************00000000          000***********00000000
00***********00000000          000***********00000000
00***********000000000          000***********00000000
000**********000000000          0000**********000000000
0000*********000000000          0000**********000000000
00000*********000000000          0000**********000000000
00000*********000000000          0000**********000000000
00000*********00000000          0000**********000000000
00000**********0000000          0000**********000000000
00000***********000000          000**********000000000
00000************000000          000***********00000000
0000*************00000          000***********00000000
00000************00000          000***********00000000
00000************00000          0000**********000000000
000000**********000000          0000**********000000000
0000000*********0000000          0000**********000000000
000000000******000000000          0000**********000000000
000000000000000000000000          00000********0000000000
000000000000000000000000          00000********0000000000
```

4e. *After 325 steps* $(Nu = 1.76)$ 4f. *After 430 steps* $(Nu = 1.77)$

The evolution of the convection is shown in Figs. 4a, 4b, 4c, 4d, 4e, and 4f. The hexagonal pattern introduced into the cell is not preserved. The system evolves through various stages, and finally settles as a roll with period $4\pi/a\sqrt{3}$. The value of Nu evaluated at the lower boundary is printed at the bottom of each figure. The steady state value for a roll is 1.76. The final configuration of the system is independent of the initial perturbation. The calculation was not pursued until a completely steady state had been achieved because that would have been excessively time consuming on the computer. It is known from previous work that steady rolls can be achieved, and that the mesh used here provides an adequate representation.

Conclusion and Applications. The Benard convection problem is not considered to be an easy problem to solve numerically even in the two-dimensional case. The fact that with our method reliable time-dependent results can be obtained even in three space dimensions indicates that the Navier-Stokes equations do indeed lend themselves to numerical solution. A number of applications to convection problems, with or without rotation, can be contemplated; in particular, it appears to be of interest to study systematically the stability of Benard convection cells when $\sigma \neq \infty$, and when the perturbations have a finite amplitude.

Other applications should include the study of the finite amplitude instability of Poiseuille flow, the stability of Couette flow, and similar problems.

New York University
Courant Institute of Mathematical Sciences
New York, New York 10012

1. H. FUJITA & T. KATO, "On the Navier-Stokes initial value problem. I," *Arch. Rational Mech. Anal.*, v. 16, 1964, pp. 269–315. MR **29** #3774.
2. A. J. CHORIN, "A numerical method for solving incompressible viscous flow problems," *J. Computational Physics*, v. 2, 1967, p. 12.
3. J. O. WILKES, "The finite difference computation of natural convection in an enclosed cavity," Ph.D. Thesis, Univ. of Michigan, Ann Arbor, Mich., 1963.
4. A. A. SAMARSKIĬ, "An efficient difference method for solving a multi-dimensional parabolic equation in an arbitrary domain," *Ž. Vyčisl. Mat. i Mat. Fiz.*, v. 2, 1962, pp. 787–811 = *U.S.S.R. Comput. Math. and Math. Phys.*, v. 1963, 1964, no. 5, pp. 894–926. MR **32** #609.
5. R. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N. J., 1962.
6. P. R. GARABEDIAN, "Estimation of the relaxation factor for small mesh size," *Math. Comp.*, v. 10, 1956, pp. 183–185. MR **19**, 583.
7. C. E. PEARSON, "A computational method for time dependent two dimensional incompressible viscous flow problems," Report No. SRRC-RR-64-17, Sperry Rand Research Center, Sudbury, Mass., 1964.
8. A. J. CHORIN, "The numerical solution of the Navier-Stokes equations for incompressible fluid," AEC Research and Development Report No. NYO-1480-82, New York Univ., Nov. 1967.
9. S. CHANDRASEKHAR, *Hydrodynamic and Hydromagnetic Stability*, Internat. Series of Monographs on Physics, Clarendon Press, Oxford, 1961. MR **23** #1270.
10. A. J. CHORIN, "Numerical study of thermal convection in a fluid layer heated from below," AEC Research and Development Report No. NYO-1480-61, New York Univ., Aug. 1966.
11. P. H. RABINOWITZ, "Nonuniqueness of rectangular solutions of the Benard problem," *Arch. Rational Mech. Anal.* (To appear.)
12. E. L. KOSCHMIEDER, "On convection on a uniformly heated plane," *Beitr. Physik. Atm.*, v. 39, 1966, p. 1.
13. H. T. ROSSBY, "Experimental study of Benard convection with and without rotation," Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Mass., 1966.
14. F. BUSSE, "On the stability of two dimensional convection in a layer heated from below," *J. Math. Phys.*, v. 46, 1967, p. 140.