SIAM J. SCI. STAT. COMPUT.
Vol. 9, No. 2, March 1988

THE ODD-EVEN HOPSCOTCH PRESSURE CORRECTION SCHEME FOR THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS*

J. H. M. TEN THIJE BOONKKAMP†

Abstract. The odd-even hopscotch (OEH) scheme is a time-integration technique for time-dependent partial differential equations. In this paper we apply the OEH scheme to the incompressible Navier–Stokes equations in conservative form. In order to decouple the computation of the velocity and the pressure, the OEH scheme is applied in combination with the pressure correction technique. The resulting scheme is referred to as the odd-even hopscotch pressure correction (OEH-PC) scheme. This scheme requires per time step the solution of a Poisson equation for the computation of the pressure. For space discretization we use standard central differences. We applied the OEH-PC scheme to the Navier–Stokes equations for the computation of an exact solution, with the purpose of testing the (order of) accuracy of the scheme in time as well as in space. Furthermore we applied the OEH-PC scheme for the computation of a model problem. Finally, a comparison between two Poisson solvers for the computation of the pressure is presented.

Key words. Navier–Stokes equations, odd-even hopscotch method, pressure correction method

AMS(MOS) subject classifications. 65M20, 76D05

1. The OEH-PC scheme: time-integration.

In this section we consider the odd-even hopscotch (OEH) scheme applied to the incompressible Navier–Stokes equations in conservative form. The OEH scheme is an integration scheme for time-dependent partial differential equations (PDEs), and it is applicable to wide classes of problems. In addition, it possesses attractive computational properties which make the scheme relatively easy to implement. For a detailed discussion of the OEH scheme the reader is referred to [5] and [6]. Application to the compressible Navier–Stokes equations of a scheme related to the OEH scheme is discussed in [16] and [17].

We adopt the pressure correction approach, which means that during the time stepping process the computation of the velocity and the pressure is decoupled in a predictor-corrector fashion. In what follows, the resulting scheme will be referred to as the odd-even hopscotch pressure correction scheme (OEH-PC scheme). A discussion of the pressure correction approach can be found in [1], [2] and [12].

Consider the incompressible Navier–Stokes equations in conservative form in d space dimensions ($d=2$ or $d=3$) [15]

$$\mathbf{u}_t=\mathbf{f}(\mathbf{u})-\nabla p, \quad \text{with } \mathbf{f}(\mathbf{u})=-\nabla\cdot(\mathbf{uu})+\frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}, \quad t>0, \quad \mathbf{x}\in\Omega, \tag{1.1}$$

$$\nabla\cdot\mathbf{u}=0, \quad t>0, \quad \mathbf{x}\in\Omega, \tag{1.2}$$

where $\mathbf{u}$ is the (scaled) velocity, p the (scaled) pressure, and Re the Reynolds number. Boundary conditions, to be specified for the velocity field $\mathbf{u}$ on the boundary Γ of the connected space domain Ω, will be introduced later. We shall present the OEH-PC scheme for (1.1), (1.2) by following the method of lines approach [11]. Thus we suppose first that by an appropriate finite difference space discretization the PDE problem (1.1), (1.2) is replaced by a system of (time-continuous) ordinary differential equations (ODEs) coupled with a set of (time-continuous) algebraic equations

$$\dot{\mathbf{U}}=\mathbf{F}(\mathbf{U})-GP, \tag{1.3}$$

$$D\mathbf{U}=B. \tag{1.4}$$

* Received by the editors October 22, 1986; accepted for publication (in revised form) May 6, 1987.
† Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, the Netherlands.

In (1.3) and (1.4), $\mathbf{F}(\mathbf{U})$ is the finite difference replacement of $\mathbf{f}(\mathbf{u})$, G and D are the finite difference replacements of the gradient- and divergence-operator, respectively, and B is a term containing boundary values for the velocity $\mathbf{u}$.

At this stage of development of the OEH-PC scheme, there is no need to be precise on the form of (1.3), (1.4). It suffices to mention that $\mathbf{U}$, $\mathbf{F}$ and P are grid functions (vectors) defined on a space grid covering Ω. G and D are (nonsquare) constant matrices and B is a vector. In what follows, $j=(j_1,\cdots,j_d)$ is a multi-index connected to the grid point $\mathbf{x}_j$ of the space grid under consideration and $\mathbf{U}_j$ the component of $\mathbf{U}$ in the $\mathbf{x}_j$-direction (and likewise for P, $\mathbf{F}$, B).

We are now ready to define the OEH-PC scheme for the semidiscrete PDE problem (1.3), (1.4). First we consider only the ODE system (1.3). (Suppose for the time being that P is a known forcing term.) For this system the OEH scheme is given by the numerical integration formula

$$\text{(1.5)}\qquad \mathbf{U}_j^{n+1}-\tau\theta_j^{n+1}(\mathbf{F}(\mathbf{U})_j^{n+1}-(GP)_j^{n+1})=\mathbf{U}_j^n+\tau\theta_j^n(\mathbf{F}(\mathbf{U})_j^n-(GP)_j^n).$$

Here $\tau=t_{n+1}-t_n$ is the time step, $\mathbf{U}_j^n$ stands for the fully discrete approximation to $\mathbf{U}_j(t_n)$, and θ is a grid function whose components θ_j^n are defined by [5], [6]

$$\text{(1.6)}\qquad \theta_j^n=\begin{cases}1 & \text{if } n+\sum_i j_i \text{ is odd (odd points)},\\ 0 & \text{if } n+\sum_i j_i \text{ is even (even points)}.\end{cases}$$

Note that if we keep n fixed, then (1.5) is just the explicit Euler rule at the odd points and the implicit Euler rule at the even ones. Alternating between the explicit and implicit Euler rules over the time-space grid is the essential feature of the OEH scheme. We return to this point later in the paper.

Writing down two successive steps of scheme (1.5) yields

$$\text{(1.7a)}\qquad \mathbf{U}_j^{n+1}=\mathbf{U}_j^n+\tau\theta_j^n\mathbf{F}(\mathbf{U})_j^n+\tau\theta_j^{n+1}\mathbf{F}(\mathbf{U})_j^{n+1}-\tau(GP)_j^n,$$

$$\text{(1.7b)}\qquad \mathbf{U}_j^{n+2}=\mathbf{U}_j^{n+1}+\tau\theta_j^{n+1}\mathbf{F}(\mathbf{U})_j^{n+1}+\tau\theta_j^{n+2}\mathbf{F}(\mathbf{U})_j^{n+2}-\tau(GP)_j^{n+2}.$$

Notice that in (1.7a) P is set at time level $t_n=n\tau$ and in (1.7b) at level $t_{n+2}=(n+2)\tau$. On a fixed space grid (1.7) may be interpreted as a second order integration formula, using stepsize 2τ, for the ODE system (1.3). A somewhat more convenient form of (1.7) is, using stepsize τ instead of 2τ,

$$\text{(1.8a)}\qquad \tilde{\mathbf{U}}=\mathbf{U}^n+\tfrac{1}{2}\tau\mathbf{F}_O(\mathbf{U}^n)+\tfrac{1}{2}\tau\mathbf{F}_E(\tilde{\mathbf{U}})-\tfrac{1}{2}\tau GP^n,$$

$$\text{(1.8b)}\qquad \mathbf{U}^{n+1}=\tilde{\mathbf{U}}+\tfrac{1}{2}\tau\mathbf{F}_E(\tilde{\mathbf{U}})+\tfrac{1}{2}\tau\mathbf{F}_O(\mathbf{U}^{n+1})-\tfrac{1}{2}\tau GP^{n+1},$$

where $\mathbf{F}_O$ is the restriction of $\mathbf{F}$ to the odd points, etc. Note that $\mathbf{F}_O+\mathbf{F}_E=\mathbf{F}$. In (1.8) $\tilde{\mathbf{U}}$ is interpreted as a result from an intermediate time level like in a Runge–Kutta formula. We shall use this formulation in the remainder of the section.

Consider (1.8a), (1.8b) coupled with the (time-discretized) set of algebraic equations

$$\text{(1.8c)}\qquad D\mathbf{U}^{n+1}=B^{n+1}.$$

The computation of $\mathbf{U}^{n+1}$ and P^{n+1} requires the simultaneous solution of (1.8b) and (1.8c). We compute on approximation to $\mathbf{U}^{n+1}$ and P^{n+1}, by following the well-known pressure correction approach [1], [2], [12], in which the computation of the velocity and pressure at the new time level is decoupled in the predictor-corrector fashion.

Substitution of P^n for P^{n+1} in (1.8b) defines the predicted velocity $\tilde{\tilde{\mathbf{U}}}$:

$$\tilde{\tilde{\mathbf{U}}} = \tilde{\mathbf{U}} + \tfrac{1}{2}\tau \mathbf{F}_E(\tilde{\mathbf{U}}) + \tfrac{1}{2}\tau \mathbf{F}_O(\tilde{\tilde{\mathbf{U}}}) - \tfrac{1}{2}\tau G P^n. \tag{1.9}$$

The corrected velocity and pressure (which we hereafter also denote by $\mathbf{U}^{n+1}$ and P^{n+1} and hence should not be mixed up with the approximations in (1.8a), (1.8b) and (1.8c)) are then defined by replacing $\mathbf{F}_O(\mathbf{U}^{n+1})$ in (1.8b) by $\mathbf{F}_O(\tilde{\tilde{\mathbf{U}}})$:

$$\mathbf{U}^{n+1} = \tilde{\mathbf{U}} + \tfrac{1}{2}\tau \mathbf{F}_E(\tilde{\mathbf{U}}) + \tfrac{1}{2}\tau \mathbf{F}_O(\tilde{\tilde{\mathbf{U}}}) - \tfrac{1}{2}\tau G P^{n+1}, \tag{1.10}$$

together with the discrete continuity equation (1.8c). From (1.9) and (1.10) we trivially obtain

$$\mathbf{U}^{n+1} - \tilde{\tilde{\mathbf{U}}} = -\tfrac{1}{2}\tau G Q^n, \qquad Q^n = P^{n+1} - P^n. \tag{1.11}$$

The trick of the pressure correction approach is now to multiply (1.11) by D and to write, using (1.8c),

$$LQ^n = \frac{2}{\tau}(D\tilde{\tilde{\mathbf{U}}} - B^{n+1}), \qquad L = DG. \tag{1.12}$$

Note that (1.12) is a Poisson equation for Q^n ($L = DG$ is a discretization of the Laplace operator $\nabla \cdot (\nabla)$). The correction Q^n for the pressure can be computed from (1.12), and once Q^n is known, the new velocity $\mathbf{U}^{n+1}$ can be directly determined from (1.11).

To sum up, the OEH-PC scheme for the semidiscrete Navier-Stokes problem (1.3), (1.4) reads

$$\tilde{\mathbf{U}} = \mathbf{U}^n + \frac{1}{2}\tau \mathbf{F}_O(\mathbf{U}^n) + \frac{1}{2}\tau \mathbf{F}_E(\tilde{\mathbf{U}}) - \frac{1}{2}\tau G P^n, \tag{1.13a}$$

$$\tilde{\tilde{\mathbf{U}}} = \tilde{\mathbf{U}} + \frac{1}{2}\tau \mathbf{F}_E(\tilde{\mathbf{U}}) + \frac{1}{2}\tau \mathbf{F}_O(\tilde{\tilde{\mathbf{U}}}) - \frac{1}{2}\tau G P^n, \tag{1.13b}$$

$$LQ^n = \frac{2}{\tau}(D\tilde{\tilde{\mathbf{U}}} - B^{n+1}), \qquad P^{n+1} = P^n + Q^n, \tag{1.13c}$$

$$\mathbf{U}^{n+1} = \tilde{\tilde{\mathbf{U}}} - \frac{1}{2}\tau G Q^n. \tag{1.13d}$$

When combined with a suitable space discretization, the OEH-PC scheme possesses various advantageous features. We shall discuss this in greater detail in the next section for symmetric finite differences on a staggered grid.

For the boundary conditions for the intermediate velocities we take the first order approximations $\tilde{\mathbf{U}} = \mathbf{u}(t_{n+1/2})$ and $\tilde{\tilde{\mathbf{U}}} = \mathbf{u}(t_{n+1})$ on Γ, where $\mathbf{u}$ is the exact boundary value for the velocity. The initial pressure P^0 is computed from the Poisson equation

$$LP^0 = D\mathbf{F}(\mathbf{U}^0) - \dot{B}^0, \tag{1.14}$$

which can be easily derived from (1.3) and (1.4).

We conclude this section with some remarks. First, the second stage (1.13b) can be economized by using its equivalent fast form (cf. [5], [6])

$$\tilde{\tilde{\mathbf{U}}}_E = 2\tilde{\mathbf{U}}_E - \mathbf{U}_E^n, \qquad \tilde{\tilde{\mathbf{U}}}_O = \tilde{\mathbf{U}}_O + \frac{1}{2}\tau \mathbf{F}_O(\tilde{\tilde{\mathbf{U}}}) - \frac{1}{2}\tau (GP^n)_O. \tag{1.13b'}$$

Our implementation is based on this fast form. Second, in the derivation of scheme (1.13) no use has been made of the particular definition of $\mathbf{F}_O$ and $\mathbf{F}_E$, except that $\mathbf{F}_O + \mathbf{F}_E = \mathbf{F}$. Consequently, in the spirit of the method of lines formulation [11], pressure correction schemes using other splittings of **F**, such as ADI, can also be described by (1.13) (see e.g., [12], where an ADI splitting is used). It is of further interest to note that when considered as a solver for the ODE system (1.3) coupled with the set (1.4), the OEH-PC scheme is of second order, for the computation of U and of first order for the computation of *P*.

Finally, the OEH-PC scheme requires roughly the same number of operations per time-step as the forward Euler scheme (we will demonstrate this in § 2.1), but has much better stability properties. To illustrate this, consider the convection-diffusion equation which models the convective and viscous effects of the Navier-Stokes equations

$$u_t+(\mathbf{q}\cdot\nabla)u=\varepsilon\nabla^2 u,\quad t>0,\quad \mathbf{x}\in\mathbb{R}^d. \tag{1.15}$$

Here $u(\mathbf{x}, t)$ represents the convected and diffused variable, the vector $\mathbf{q}=(q_1,\cdots,q_d)^T$ the (constant) velocity, and $\varepsilon>0$ a viscosity parameter. Suppose that for space discretization we use standard central differences, with constant grid size h in all space-directions. If the OEH scheme is formulated like in (1.8), then von Neumann stability analysis applied to this scheme yields the following necessary and sufficient time step restriction [21]

$$d\left(\frac{\tau}{h}\right)^2\sum_{k=1}^{d} q_k^2\leqq 4. \tag{1.16}$$

For the forward Euler-central difference scheme, the time step restrictions for von Neumann stability are [10], [21]

$$\frac{2d\varepsilon\tau}{h^2}\leqq 1,\qquad \sum_{k=1}^{d}\frac{q_k^2\tau}{2\varepsilon}\leqq 1. \tag{1.17}$$

The second inequality of (1.17) (convection-diffusion barrier) shows that the forward Euler-central difference scheme becomes unconditionally unstable as $\varepsilon\to 0$, whereas the OEH scheme is conditionally stable uniformly in ε, i.e., $\tau=O(h)$ independent of ε. Observe that the first inequality for the forward Euler-central difference scheme implies $\tau=O(\varepsilon^{-1}h^2)$, which is disadvantageous for larger values of ε. It is fair to say that, in general, a disadvantage of the OEH-central difference scheme is the so-called Du Fort-Frankel deficiency [5], [21]. However, as we will point out in § 3.2, in the present application this disadvantage is of minor importance.

2. The OEH-PC scheme: space discretization. In § 2.1 we will discuss the space discretization on a staggered grid of the Navier-Stokes problem, which defines the fully discrete OEH-PC scheme. We will show that due to the conservative form, our fully discrete OEH-PC scheme is in fact an explicit scheme, which needs only one array of storage for the computation of the velocity. In § 2.2 we will discuss the Poisson equation for the pressure, and in § 2.3 we will discuss (briefly) the space discretization on two other grids. For the sake of presentation, we restrict ourselves to two-dimensional rectangular domains.

2.1. Space discretization on a staggered grid. Consider the two-dimensional incompressible Navier-Stokes equations in conservative form

$$u_t=f_1(u,v)-p_x\quad \text{with } f_1(u,v)=-(u^2)_x-(uv)_y+\frac{1}{\mathrm{Re}}(u_{xx}+u_{yy}), \tag{2.1a}$$

$$v_t=f_2(u,v)-p_y\quad \text{with } f_2(u,v)=-(uv)_x-(v^2)_y+\frac{1}{\mathrm{Re}}(v_{xx}+v_{yy}), \tag{2.1b}$$

$$u_x+v_y=0, \tag{2.2}$$

with boundary conditions

(2.3) $$u = u_\Gamma, \qquad v = v_\Gamma \quad \text{on } \Gamma = \partial\Omega.$$

Note that there are no pressure boundary conditions available, although we have to solve a Poisson equation for the pressure. We will return to this point later in the section.

For the space discretization, we use the staggered grid first introduced by Harlow and Welch [8], see Fig. 1. The application of standard, second order central differences on this grid converts (2.1a) and (2.1b) into (cf. (1.3))

(2.4a) $$\dot{U}_{ij} = F_{1,ij}(U, V) - d_x P_{ij}, \qquad i = 1(1)N-1, \quad j = 1(1)M \quad (\text{interior} \times\text{-points}),$$

(2.4b) $$\dot{V}_{ij} = F_{2,ij}(U, V) - d_y P_{ij}, \qquad i = 1(1)N, \quad j = 1(1)M-1 \quad (\text{interior} \bigcirc\text{-points}),$$

where

(2.5a) $$\begin{aligned} F_{1,ij}(U, V) = &-\frac{1}{2h}(U^2_{i+1,j} - U^2_{i-1,j}) - \frac{1}{2k}(U_{i,j+1}\bar{V}_{i,j+1} - U_{i,j-1}\bar{V}_{i,j-1}) \\ &+ \frac{1}{\text{Re}\, h^2} \cdot (U_{i+1,j} - 2U_{ij} + U_{i-1,j}) + \frac{1}{\text{Re}\, k^2}(U_{i,j+1} - 2U_{ij} + U_{i,j-1}), \end{aligned}$$

(2.5b) $$\begin{aligned} F_{2,ij}(U, V) = &-\frac{1}{2h}(\bar{U}_{i+1,j}V_{i+1,j} - \bar{U}_{i-1,j}V_{i-1,j}) - \frac{1}{2k}(V^2_{i,j+1} - V^2_{i,j-1}) \\ &+ \frac{1}{\text{Re}\, h^2} \cdot (V_{i+1,j} - 2V_{ij} + V_{i-1,j}) + \frac{1}{\text{Re}\, k^2} \cdot (V_{i,j+1} - 2V_{ij} + V_{i,j-1}), \end{aligned}$$

(2.5c) $$d_x P_{ij} = \frac{1}{h}(P_{i+1,j} - P_{ij}),$$

(2.5d) $$d_y P_{ij} = \frac{1}{k}(P_{i,j+1} - P_{ij}).$$

Note that in the above formulation U, V and P are time-continuous grid functions whose components U_{ij}, V_{ij} and P_{ij} approximate the velocities u, v and the pressure p,

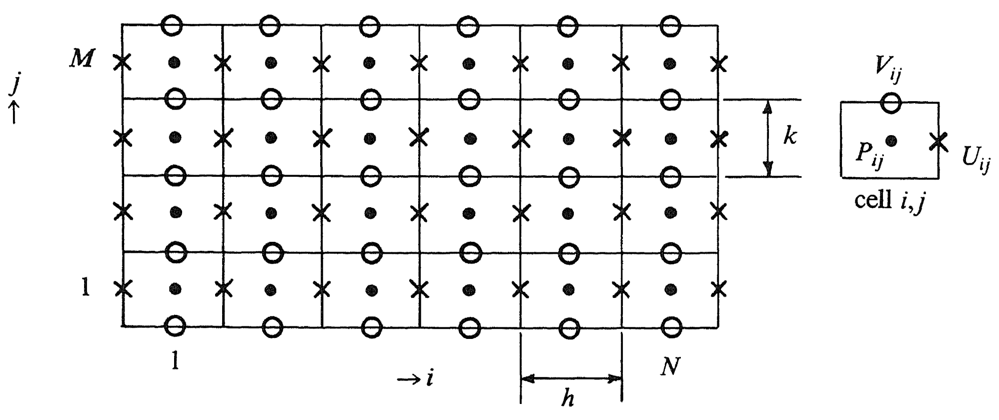

FIG. 1. *The staggered grid.*

respectively, at the corresponding gridpoints. In (2.5a) $\bar{V}_{ij}$ represents an approximation to V in the ×-points (points where U is defined); likewise $\bar{U}_{ij}$ represents an approximation to U in the ○-points. The values of $\bar{V}_{ij}$ and $\bar{U}_{ij}$ are determined by averaging over neighbouring values of V_{ij} and U_{ij} respectively, in such a way that the odd-even coupling between the variables is preserved. This means that a variable in an odd point is only coupled with variables in even points and vice versa. This leads to

$$\bar{U}_{ij} = \tfrac{1}{2}(U_{ij} + U_{i-1,j+1}), \qquad \bar{V}_{ij} = \tfrac{1}{2}(V_{ij} + V_{i+1,j-1}). \tag{2.6}$$

The space discretization of (2.1), as defined in (2.4), (2.5) determines the vector-function $\mathbf{F}(\mathbf{U})$ and the operator G in (1.3). Let $\mathbf{U} = (U, V)^T$, then $\mathbf{F}_{ij}(\mathbf{U}) = (F_{1,ij}(U, V), F_{2,ij}(U, V))^T$ and $GP_{ij} = (d_x P_{ij}, d_y P_{ij})^T$.

Concerning the boundary conditions for the velocity we note the following. Consider, for example, (2.1a) in the ×-points $(i, 1)(i = 1(1)N-1)$. Discretization of the derivatives $(uv)_y$ and u_{yy} would require values outside the computational domain. Therefore we replace the central difference approximations to $(uv)_y$ and u_{yy} by the following noncentered first order differences [15], which preserve the odd-even coupling between the variables

$$((uv)_y)_{i1} = \frac{2}{3k}(U_{i2}\bar{V}_{i2} - u(ih, 0)v(ih, 0)), \tag{2.7a}$$

$$(u_{yy})_{i1} = \frac{4}{3k^2}(U_{i2} - 3U_{i1} + 2u(ih, 0)). \tag{2.7b}$$

Second order noncentered approximations to $(uv)_y$ and u_{yy} would destroy the odd-even coupling.

Space discretization of (2.2) in all ·-points (using central differences) yields

$$(D\mathbf{U})_{ij} := \frac{1}{h}(U_{ij} - U_{i-1,j} + \beta(V_{ij} - V_{i,j-1})) = 0, \tag{2.8}$$

where $\beta = h/k$. Note that boundary values for U or V occurring in (2.8) are written in the right-hand side B (cf. (1.4)). For example, for $j = 1$, (2.2) is discretized as

$$(D\mathbf{U})_{i1} := \frac{1}{h}(U_{i1} - U_{i-1,1} + \beta V_{i1}) = B_{i1} = \frac{1}{k} V_{i0}. \tag{2.8'}$$

Having defined the operators G and D, one can easily deduce the following expression for the operator L

$$\begin{aligned}(LQ)_{ij} = D(GQ)_{ij} &= \frac{1}{h}(d_x Q_{ij} - d_x Q_{i-1,j} + \beta(d_y Q_{ij} - d_y Q_{i,j-1}))\\ &= \frac{1}{h^2}(\beta^2 Q_{i,j-1} + Q_{i-1,j} - (2+2\beta^2)Q_{ij} + Q_{i+1,j} + \beta^2 Q_{i,j+1}),\end{aligned} \tag{2.9}$$

which is the standard 5-point molecule for the Laplace operator. Near a boundary (2.9) takes a different form, because of the different definition of the operator D. For example for $j = 1$, one finds

$$\begin{aligned}(LQ)_{i1} = D(GQ)_{i1} &= \frac{1}{h}(d_x Q_{i1} - d_x Q_{i-1,1} + \beta d_y Q_{i1})\\ &= \frac{1}{h^2}(Q_{i-1,1} - (2+\beta^2)Q_{i1} + Q_{i+1,1} + \beta^2 Q_{i2}).\end{aligned} \tag{2.9'}$$

Now, consider (1.13c) at the $\cdot$-points $(i, 1)(i = 1(1)N)$. Using (2.8), (2.8′), (2.9) and (2.9′), it is easy to see that $(Q^n_{i0} - Q^n_{i1})/k = 2(V^{n+1}_{i0} - \tilde{V}_{i0})/\tau = 0$, which is the (central difference) approximation of $(\partial Q^n/\partial n)_{i0} = 0$, where n is the outward unit normal on $x = 0$. A similar argument applied to the Poisson equation for the initial pressure P^0 ((1.14)) leads for $j = 1$ to the boundary condition $(\partial P^0/\partial n)_{i0} = \dot{V}^0_{i0} - F_{2i0}(U^0, V^0)$, which is in accordance with the Navier-Stokes equations. Hence we see that a Neumann condition for the pressure (-increment) is automatically involved in the scheme.

Thus the scheme implies $\partial P^n/\partial n = \partial P^0/\partial n$ on Γ at every time level $t_n = n\tau$, although the exact pressure does not in general satisfy this condition. Most methods involve artificial pressure boundary conditions, like, for example, the projection method [2], [4], [15], [20]. In [20] Temam defines a projection method, which is a predictor corrector method like the pressure correction method, in which in the predictor step the pressure term is completely omitted. This scheme implies the "unphysical" boundary condition $\partial P/\partial n = 0$. Nevertheless, he proves the convergence of his scheme. Therefore, it is believed that our OEH-PC scheme does converge too, although the artificial pressure condition will lead to some loss of accuracy. This is demonstrated with a numerical example in § 3.1. A proof of convergence is out of the scope of the present paper.

Having defined the space discretization, we now discuss in some detail the merits of the resulting fully discrete OEH-PC scheme. Consider (1.13a) and (1.13b) of the OEH-PC scheme. The order of computation is

(2.10a) $$\tilde{\mathbf{U}}_O = \mathbf{U}^n_O + \tfrac{1}{2}\tau\mathbf{F}_O(\mathbf{U}^n) - \tfrac{1}{2}\tau(GP^n)_O,$$

(2.10b) $$\tilde{\mathbf{U}}_E = \mathbf{U}^n_E + \tfrac{1}{2}\tau\mathbf{F}_E(\tilde{\mathbf{U}}) - \tfrac{1}{2}\tau(GP^n)_E,$$

(2.10c) $$\tilde{\tilde{\mathbf{U}}}_E = \tilde{\mathbf{U}}_E + \tfrac{1}{2}\tau\mathbf{F}_E(\tilde{\mathbf{U}}) - \tfrac{1}{2}\tau(GP^n)_E = 2\tilde{\mathbf{U}}_E - \mathbf{U}^n_E,$$

(2.10d) $$\tilde{\tilde{\mathbf{U}}}_O = \tilde{\mathbf{U}}_O + \tfrac{1}{2}\tau\mathbf{F}_O(\tilde{\tilde{\mathbf{U}}}) - \tfrac{1}{2}\tau(GP^n)_O.$$

This scheme is in fact an explicit scheme. To demonstrate this, consider the computation of $\tilde{\mathbf{U}}$. Clearly the computation of $\tilde{\mathbf{U}}_O$ is explicit. Equation (2.10b) for the computation of $\tilde{\mathbf{U}}_E$ reads for the U-component in an even point (i, j) (substitute (2.5a), (2.5c) and (2.6))

(2.11)
$$\begin{aligned}\tilde{U}_{ij} = U^n_{ij} &- \frac{\tau}{4h}(\tilde{U}^2_{i+1,j} - \tilde{U}^2_{i-1,j}) - \frac{\tau}{8k}(\tilde{U}_{i,j+1}(\tilde{V}_{i,j+1} + \tilde{V}_{i+1,j}) \\ &- \tilde{U}_{i,j-1}(\tilde{V}_{i,j-1} + \tilde{V}_{i+1,j-2})) + \frac{\tau}{2\,\mathrm{Re}\,h^2}(\tilde{U}_{i+1,j} - 2\tilde{U}_{ij} + \tilde{U}_{i-1,j}) \\ &+ \frac{\tau}{2\,\mathrm{Re}\,k^2}(\tilde{U}_{i,j+1} - 2\tilde{U}_{ij} + \tilde{U}_{i,j-1}) - \frac{\tau}{2h}(P^n_{i+1,j} - P^n_{ij}).\end{aligned}$$

The values of $\tilde{U}_{i\pm1,j}$, $\tilde{U}_{i,j\pm1}$, $\tilde{V}_{i,j\pm1}$ and $\tilde{V}_{i+1,j-2}$ are odd numbered values which were already computed with (2.10a). This means that (2.11) is only diagonally implicit, since $\tilde{U}_{ij}$ is the only unknown, and hence explicit. In the same way, the computation of $\tilde{V}_E$ is explicit. A similar argument applies to the computation of $\tilde{\tilde{\mathbf{U}}}$.

In scheme (2.10a)–(2.10d) the steps (2.10b) and (2.10c) are considered as one computational step: first compute $\tilde{\mathbf{U}}_E$ in a point, store this value in a dummy-variable, and then compute $\tilde{\tilde{\mathbf{U}}}_E$ in the same point, using the fast form. Taking this into consideration, we easily see that only one array of storage is required for the computation of $\tilde{\tilde{\mathbf{U}}}$, which is especially advantageous for multidimensional problems.

2.2. The Poisson equation for the pressure. The pressure increment Q^n is computed from (1.13c), where the operator L is defined as in (2.9) and (2.9′). In fact

L is the 5-point discretization of the Laplace operator with Neumann boundary conditions. Considered as a matrix, L has a few attractive properties, such as symmetry, negative definiteness and a pentadiagonal structure. There are many methods available for the solution of a set of equations with matrix L. Since the OEH scheme is very cheap per step, it is essential that we combine it with a fast Poisson solver in order to obtain a fast OEH-PC scheme. In our computations, we used the incomplete Choleski conjugate gradient (ICCG) method [13], [14] and a multigrid (MG) method [9], [19]. A comparison between these two methods will be presented in § 3.3.

2.3. Space discretizations on other grids. For the space discretization, one can also use the ordinary grid or the half-staggered grid, see Fig. 2; cf. [15]. In the ordinary grid, the components of the velocity and the pressure are all defined at the nodes of the grid. The advantage of this grid is its simplicity, especially treatment of the boundary conditions for the velocity is straightforward.

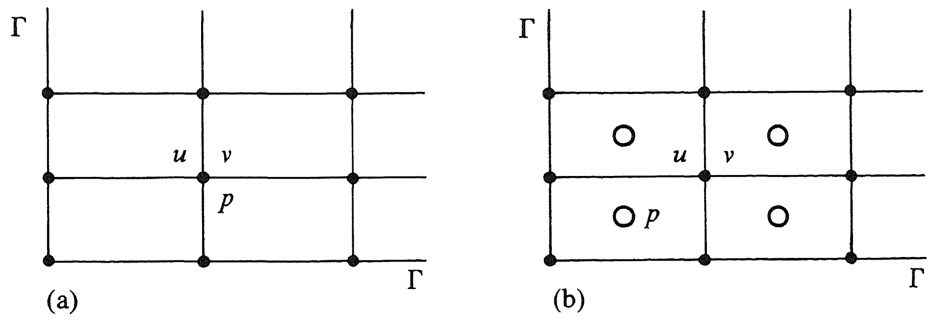

FIG. 2. *The ordinary grid* (a) *and the half-staggered grid* (b).

However a disadvantage of this grid is the fact that the pressure is defined at nodes on the boundary. Therefore, computation of the pressure in a pressure correction fashion requires pressure boundary conditions, which are generally not available. In the half-staggered grid, the components of the velocity are defined at the nodes of the grid and the pressure is defined at the centre of each cell of the grid, cf. [4], [15]. The pressure is not prescribed on the boundary anymore, and hence no pressure boundary conditions are required. A disadvantage of this grid is the fact that the discretization of the gradient- and divergence-operator is slightly more difficult than on the ordinary grid or on the staggered grid.

The major drawback of the ordinary grid and the half-staggered grid is the fact that these grids are not suitable for the computation of the pressure in a pressure correction fashion. To make this plausible, consider the molecule for the operator $L = DG$ on the ordinary grid and the half-staggered grid, respectively, when standard central differences are used for the discretization of the gradient- and divergence-operator; see Fig. 3. The operator L on the ordinary grid is again the usual 5-point discretization of the Laplacian, but now on a grid with double gridsize. The consequence is that there exist four uncoupled networks of pressure points (see Fig. 3). This leads to the existence of four independent solutions for the pressure, which differ from each other by arbitrary constants. Furthermore, due to the double gridsize, the pressure on the ordinary grid will be less accurate than on the staggered grid. The operator L on the half-staggered grid is a 9-point discretization of the Laplacian unless $\beta = 1 (h = k)$, then L is a 5-point discretization of the Laplacian denoted by the solid lines. In the latter case, the pressure field is decoupled in two independent pressure fields, which

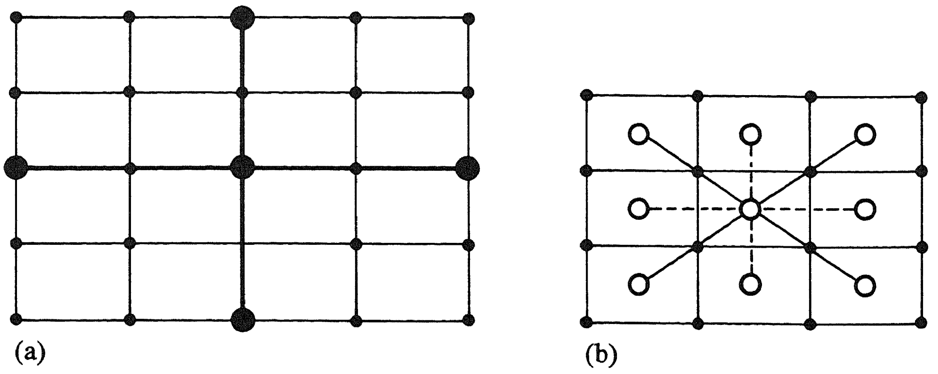

FIG. 3. *Molecule of the operator L, on the ordinary grid* (a) *and on the half-staggered grid* (b).

differ from each other by an arbitrary constant. Because of this decoupling, the ordinary grid and the half-staggered grid are not suitable for the computation of the pressure using a pressure correction scheme. However, the pressure gradient is not affected by this decoupling, and therefore one can still use these grids for the computation of the velocity. In § 3.1 we will present a numerical illustration which clearly favours the staggered grid.

3. Numerical examples. Combined with the ICCG method and an MG method for the solution of the Poisson equation, we have applied our OEH-PC scheme to two Navier–Stokes problems. The first is a simple test problem, of which the exact solution is known [2]. We used this problem to test the accuracy and the order of accuracy of the OEH-PC scheme, in time and in space (see § 3.1). Our second problem is a model problem that comes closer to practical applications. It concerns the flow through a reservoir [12] (see § 3.2). In § 3.3 we will present a comparison, based on our experiences, between the two Poisson solvers.

3.1. Accuracy and order test. In this section we discuss results of the OEH-PC scheme applied to the incompressible Navier–Stokes problem with the exact solution

$$\begin{aligned} u(x,y,t) &= -\cos\lambda(x-a)\cdot\sin\lambda(y-a)\cdot e^{-2\lambda^2 t/\mathrm{Re}},\\ v(x,y,t) &= \sin\lambda(x-a)\cdot\cos\lambda(y-a)\cdot e^{-2\lambda^2 t/\mathrm{Re}},\\ p(x,y,t) &= -\tfrac{1}{4}\cdot(\cos 2\lambda(x-a)+\cos 2\lambda(y-a))\cdot e^{-4\lambda^2 t/\mathrm{Re}}. \end{aligned} \tag{3.1}$$

In our computation we prescribed Dirichlet boundary conditions for u, v and for the parameters λ, a and Re we took: $\lambda=\pi$, $a=0$, 0.25 and $\mathrm{Re}=100$. The velocity field and the isobars for these values of λ, a and Re are displayed in Fig. 4. The computational domain is $\Omega=(0,1)\times(0,1)$ and the time-integration interval is $[0,1]$. While referring to our comments on the artificial pressure boundary condition, we notice that for $a=0$, $\partial p/\partial n=0$ on the boundary Γ and for $a=0.25$, $\partial p/\partial n\neq 0$ and a function of t on Γ. Computations were performed on a staggered grid as well as on an ordinary grid, with grid sizes $h=k=\frac{1}{10}$, $\frac{1}{20}$, $\frac{1}{40}$ and stepsizes $\tau=\frac{1}{10}$, $\frac{1}{20},\cdots,\frac{1}{160}(\tau\leqq h)$. Since $\max(u(x,y,t))=1$ and $\max(v(x,y,t))=1$, the critical time step for von Neumann stability for the related convection-diffusion equation is $\tau=h$ (cf. (1.16)).

With the purpose of testing the (order of) accuracy of the OEH-PC scheme in time, as well as in space, we compare the numerical solution to the exact solution (3.1). Let $\varepsilon_f(h,\tau)$ be the l_1-norm of the absolute error in $f(f=u, v$ or $p)$ at $t=1$, obtained for gridsize $h=k$ and stepsize τ. Then the number of significant

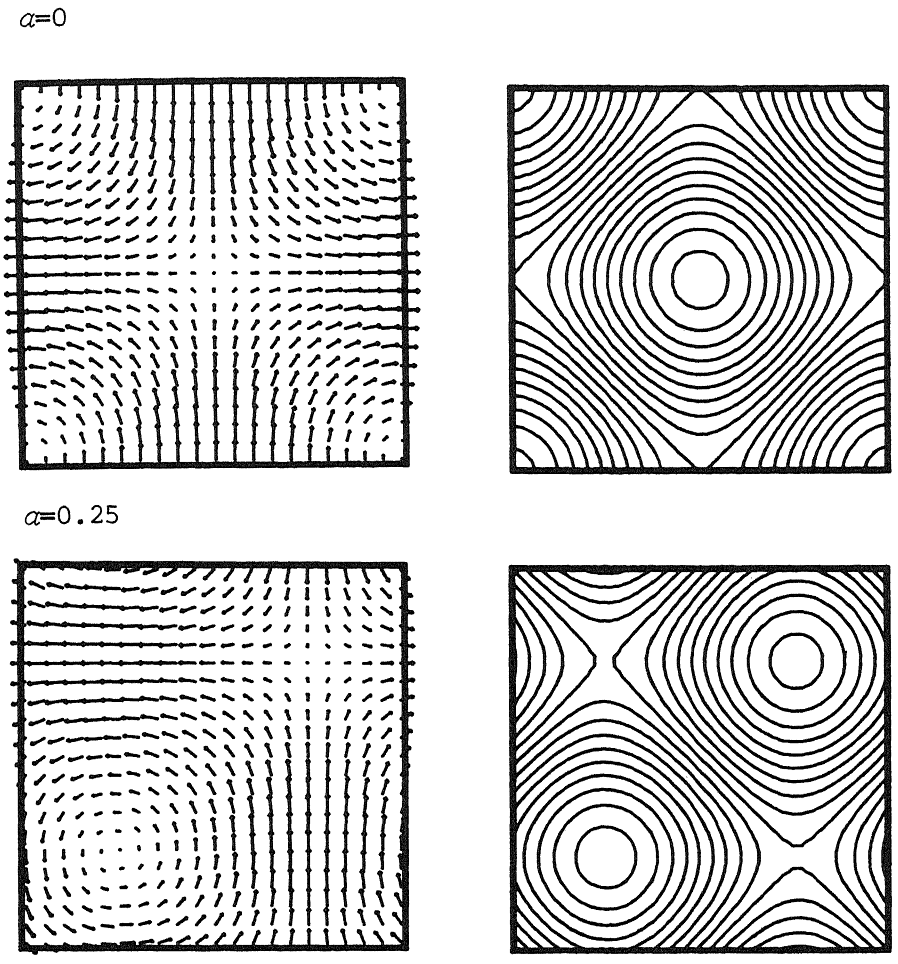

FIG. 4. *Velocity field and isobars.*

digits in f, $\lambda_f(h, \tau)$, is defined as: $\lambda_f(h, \tau) := -\log_{10}(\varepsilon_f(h, \tau))$. Table 1 displays $\lambda_u(h, \tau)$, $\lambda_v(h, \tau)$ and $\lambda_p(h, \tau)$ for the numerical solution computed on the staggered grid. For $a = 0$, when looking along rows (τ fixed, $h \to 0$), one can observe second order behaviour in space ($\log_{10}(4) \approx 0.6$), and when looking along diagonals (τ/h fixed, $\tau \to 0$), one observes second order behaviour in time and space of the OEH-PC scheme. Note that the error in the solution is dominated by the space error. In the same way, one observes that for $a = 0.25$ the velocity components u and v behave second order in space and time, and the pressure p at least first order. Comparing both solutions, we see that the solution for $a = 0.25$ is less accurate than the solution for $a = 0$, this due to the artificial pressure boundary condition for $a = 0.25$ (see § 2.1). However, the OEH-PC scheme clearly does converge for $a = 0.25$.

The same computations were performed on the ordinary grid, the results of which can be found in Table 2 (we only present results for the velocity for $a = 0$). The same conclusions concerning the order of accuracy of the OEH-PC scheme apply to this case. Comparing the results on the ordinary grid and the staggered grid, one sees that the velocity on the staggered grid is approximately ten times more accurate than on the ordinary grid. The reason for this is the inaccurate computation of the pressure (-gradient) on the ordinary grid. This clearly demonstrates that the staggered grid is

TABLE 1
$\lambda_u(h,\tau)$, $\lambda_v(h,\tau)$ *and* $\lambda_p(h,\tau)$ *for the staggered grid.*

$a = 0.0$

h^{-1} / τ^{-1}	$\lambda_u(h,\tau)$ 10	20	40	h^{-1} / τ^{-1}	$\lambda_v(h,\tau)$ 10	20	40	h^{-1} / τ^{-1}	$\lambda_p(h,\tau)$ 10	20	40
10	2.53			10	2.47			10	1.93		
20	2.53	3.22		20	2.47	3.16		20	1.91	2.63	
40	2.53	3.23	3.85	40	2.47	3.16	3.79	40	1.91	2.56	3.45
80	2.53	3.23	3.86	80	2.47	3.15	3.79	80	1.91	2.55	3.26
160	2.53	3.23	3.86	160	2.47	3.15	3.79	160	1.91	2.54	3.19

$a = 0.25$

h^{-1} / τ^{-1}	$\lambda_u(h,\tau)$ 10	20	40	h^{-1} / τ^{-1}	$\lambda_v(h,\tau)$ 10	20	40	h^{-1} / τ^{-1}	$\lambda_p(h,\tau)$ 10	20	40
10	2.01			10	1.81			10	1.72		
20	2.01	2.79		20	1.81	2.60		20	1.74	2.09	
40	2.01	2.79	3.50	40	1.81	2.60	3.36	40	1.75	2.11	2.62
80	2.01	2.79	3.50	80	1.81	2.60	3.35	80	1.75	2.12	2.64
160	2.01	2.79	3.50	160	1.81	2.60	3.35	160	1.75	2.12	2.65

TABLE 2
$\lambda_u(h,\tau)$ *and* $\lambda_v(h,\tau)$ *for the ordinary grid,* $a = 0$.

h^{-1} / τ^{-1}	$\lambda_u(h,\tau)$ 10	20	40	h^{-1} / τ^{-1}	$\lambda_v(h,\tau)$ 10	20	40
10	1.52			10	1.52		
20	1.52	2.19		20	1.52	2.20	
40	1.52	2.19	2.81	40	1.52	2.20	2.82
80	1.52	2.19	2.81	80	1.52	2.20	2.82
160	1.52	2.19	2.81	160	1.52	2.20	2.82

to be preferred to the ordinary grid, when solving the Navier–Stokes equations in a pressure correction fashion.

In order to test the accuracy of the OEH-PC scheme when considered purely as a time-integrator, it is convenient to compare the numerical solution to the exact solution of the system of ODEs which results after space discretization. As an approximation to this exact solution, we take the numerical solution computed with stepsize $\tau = 1/1280$. The l_1-norm of the absolute time error, $\varepsilon_f^*(h,\tau)$, is defined with respect to this solution, and $\lambda_f^*(h,\tau) := -\log_{10}(\varepsilon_f^*(h,\tau))$ ($f = u$, v or p). We only present results on the staggered grid for $a = 0.25$, which can be found in Table 3. The experiment clearly demonstrates the second order behaviour of the OEH-PC scheme for the computation of the velocity, when considered as an ODE time-integrator. For the computation of the pressure, the OEH-PC scheme is first order in time, though this is not quite clear for $h = 1/20$, $1/40$. This is probably due to the fact that for these values

TABLE 3
$\lambda_u^*(h,\tau)$, $\lambda_v^*(h,\tau)$ and $\lambda_p^*(h,\tau)$ for the staggered grid, $a=0.25$.

$\lambda_u^*(h,\tau)$				$\lambda_v^*(h,\tau)$				$\lambda_p^*(h,\tau)$			
h^{-1} / τ^{-1}	10	20	40	h^{-1} / τ^{-1}	10	20	40	h^{-1} / τ^{-1}	10	20	40
10	3.20			10	3.23			10	2.60		
20	3.90	3.97		20	3.99	4.03		20	3.31	3.27	
40	4.59	4.51	4.68	40	4.59	4.57	4.74	40	3.74	3.79	3.59
80	5.10	5.08	5.15	80	5.19	5.14	5.25	80	4.09	4.26	4.16
160	5.70	5.67	5.68	160	5.79	5.73	5.78	160	4.42	4.70	4.74

the asymptotics still do not hold (τ too large). We emphasize that columnwise the number of digits found correspond to different ODE systems.

Next we discuss briefly the DuFort-Frankel (DFF) deficiency [6], [21]. Consider the convection-diffusion equation (1.15), which models the convective and viscous effects of the Navier-Stokes equations. The OEH scheme for this equation is equivalent to the leapfrog-DFF scheme at the odd points, cf. [21]. Let H_k be the central difference approximation to the first space derivative in the kth direction and μ_k the standard averaging operator in the kth direction, then the leapfrog-DFF scheme for problem (1.15) reads

$$(1+2d\sigma)U_j^{n+2}=(1-2d\sigma)U_j^n-\sum_{k=1}^{d}(c_kH_k-4\sigma\mu_k)U_j^{n+1}, \tag{3.2}$$

where $\sigma=\varepsilon\tau/h^2$, $c_k=q_k\tau/h$ and h is the constant gridsize in all space directions. By the DFF deficiency we mean that for τ, $h\to 0$ the solution of scheme (3.2) will converge to the solution of the problem

$$u_t+(\mathbf{q}\cdot\nabla)u=\varepsilon\nabla^2u-\varepsilon d\left(\frac{\tau}{h}\right)^2u_{tt}. \tag{3.3}$$

In general, for convergence it is thus necessary that $\tau=o(h)$. Through the equivalence property, the same conclusion is valid for the OEH scheme. The DFF deficiency also exists for the nonlinear Burgers equation, although the equivalence to the leapfrog-DFF scheme cannot be derived in this case. Experiments in [21] showed that the OEH scheme applied to the nonlinear Burgers equation failed to converge for a fixed ratio τ/h when τ, $h\to 0$. In our example, however, the OEH scheme does not suffer from this deficiency. The reason for this is that the term εdu_{tt} is very small, and hence the DFF deficiency is practically absent. In general the DFF deficiency will have some negative influence on the accuracy. Fortunately, there is clear practical evidence (see also [21]) that in most cases this will be of only minor importance.

3.2. Flow through a reservoir. In this section we discuss results of the OEH-PC scheme when used to compute the flow in a reservoir [12] (see Fig. 5). Computations were performed subject to the following initial conditions and boundary conditions:

initial conditions: $u=v=0$ for $t=0$
boundary conditions:
no slip: $u=0$, $v=0$

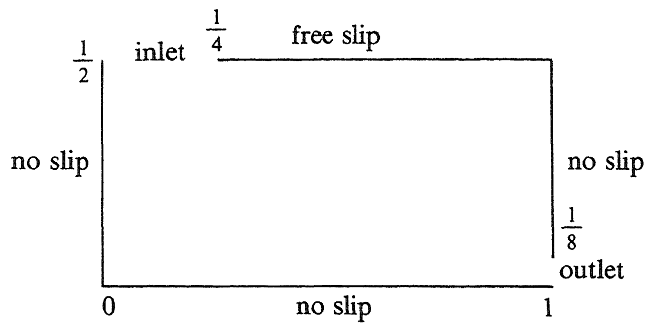

FIG. 5. *The reservoir.*

free slip: $u_y = 0,\ v = 0$
inlet: $u = 0,\ v = -432(x - \frac{1}{4})^2 x(1 - e^{-t})$
outlet: $u = 432(\frac{1}{8} - y)y(1 - e^{-t}),\ v = 0.$

Notice that the boundary conditions satisfy

$$\int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} ds = \iint_{\Omega} \nabla \cdot \mathbf{u}\, dS = 0,$$

where $\mathbf{n}$ is the unit normal on $\partial\Omega$ (conservation of mass). The outlet boundary condition, which is a Poisseuille profile, is not very realistic, especially not for high Re-numbers since it causes an artificial numerical boundary layer at the outlet. This boundary layer may cause oscillations in the solution in the interior domain [18]. Therefore, we have to look for other outlet boundary conditions with minimal influence on the interior flow field. A very suitable outlet boundary condition is the so-called traction-free boundary condition. This means that there are no viscous normal and tangential stresses at the outlet (cf. [7]), i.e.,

$$\tau_{xx} = -p + \frac{2}{\text{Re}} u_x = 0, \qquad \tau_{xy} = \frac{1}{\text{Re}}(u_y + v_x) = 0. \tag{3.4}$$

However, these boundary conditions do not easily fit in the OEH-PC scheme. Another possibility we adopt is to extend the computational domain with a horizontal pipe connected at the outlet (extended domain). The assumption here is that the flow has fully developed into a Poisseuille flow at the end of the pipe, which is a realistic assumption, provided the pipe is long enough. In our computations we did not bother about the length of the pipe, and took it equal to 1. The horizontal walls of the pipe are no-slip walls.

We have computed the solution for = 100(100)800 on the original domain as well as on the extended domain, on a staggered grid with gridsize $h = k = 1/32$. Time-integration was performed from $t = 0$ to $t = 4$. The time step τ was bounded by the linearized stability restriction $\tau/h \leqq \sqrt{2}/u_{\max}$, where $u_{\max}$ is the (modulus of the) maximum velocity (cf. (1.16)). Consequently we have chosen $\tau = \frac{1}{4}h$ for Re = 100(100)700 and $\tau = \frac{1}{8}h$ for Re = 800, although these values for τ are not the optimal ones. However, especially for increasing Re, we prefer to remain on the safe side in order to prevent nonlinear instabilities. Another reason to be careful is the fact that we use the pressure correction method, the influence of which on stability is not yet fully clear. The Poisson solver we used is the MG algorithm MGD5V (see § 3.3). Figs. 6 and 7 present the velocity and the isobars for, respectively, Re = 100, 500 and

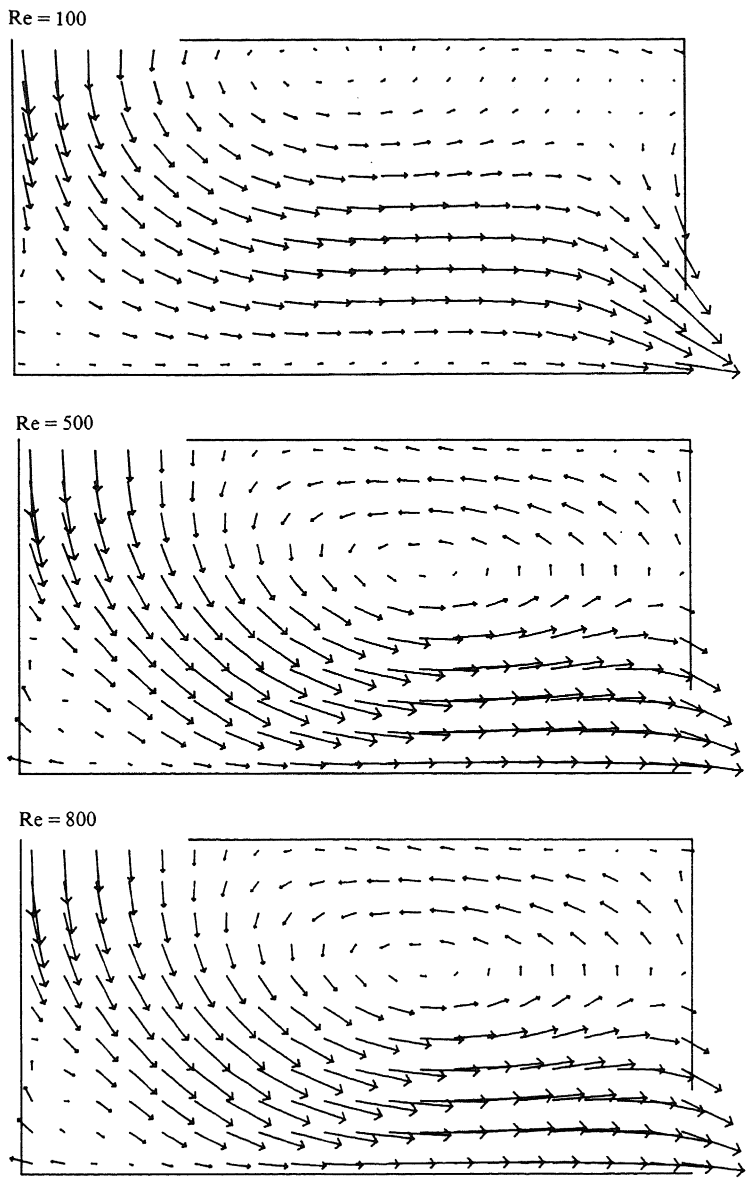

FIG. 6. *Velocity field at* $t = 4$ *for* Re = 100, 500 *and* 800.

800 at $t = 4$ computed on the extended domain (the pipe of the extended domain is not shown in these figures).

From our numerical experiments we can draw the following conclusions. For small Re-numbers (Re $\leqq$ 200), there is hardly any difference between the velocity field and the isobars computed on the original domain and on the extended domain. The velocity fields computed on both domains are almost free of oscillations. However, oscillations do occur in the velocity field for Re > 200. In this case, the velocity field computed on the extended domain is slightly better (smaller oscillations) than the velocity field computed on the original domain. The isobars computed on the original

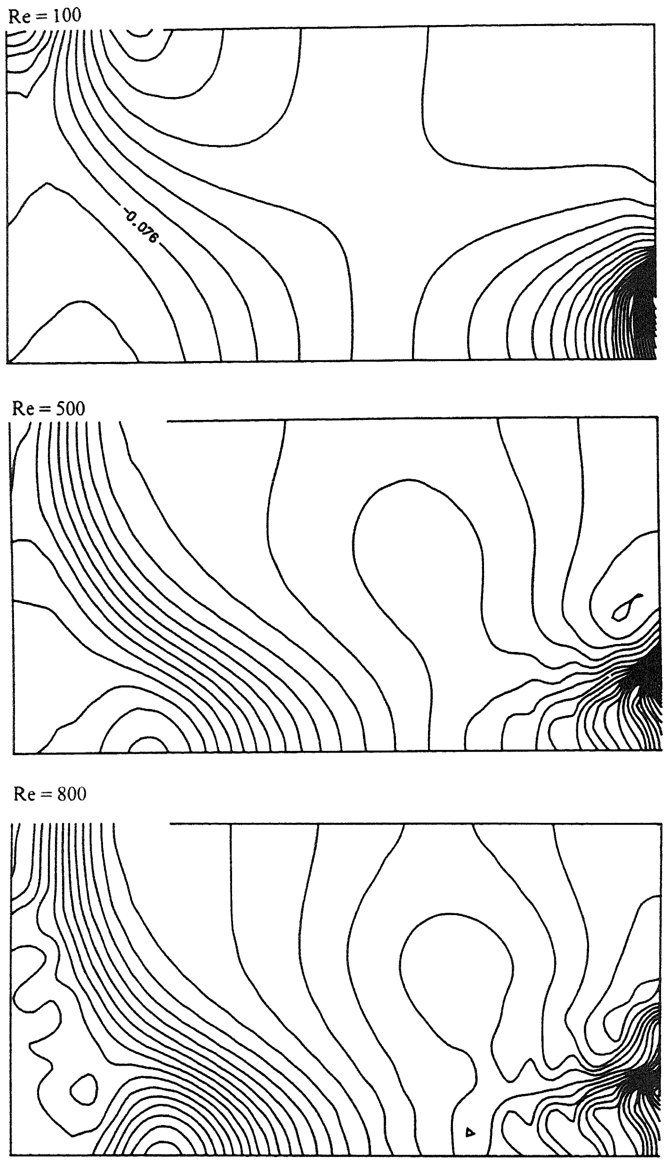

FIG. 7. *Isobars at* $t = 4$ *for* $\mathrm{Re} = 100$, 500 *and* 800.

domain for $\mathrm{Re} > 200$ are not correct, whereas the isobars computed on the extended domain are much more realistic.

We borrowed this model problem from van Kan [12]. He computes the flow (without pipe) using a pressure correction Crank-Nicolson ADI scheme (ADI-PC scheme). The outflow boundary conditions he uses are a Poisseuille profile and the traction-free boundary conditions. Comparing his results with ours, we can conclude the following. Our velocity fields are in good agreement with the corresponding ones computed by van Kan. However, his results are more disturbed by oscillations than ours, and this is due to the numerical boundary layer at the outlet occurring in his computations. We note that for the corresponding linear convection-diffusion problem (1.15) the ADI scheme is unconditionally stable, whereas the OEH scheme is only conditionally stable. In practice the ADI scheme is often only conditionally stable, especially for high Re-numbers, because of the nonlinearity of the convective terms [12]. Nevertheless, the ADI scheme possesses better stability properties than the OEH scheme, so that with respect to stability he normally can take larger time steps. The computational costs per time step for the OEH scheme are less than those for the ADI scheme, since the OEH scheme is in fact an explicit scheme and the ADI scheme requires the solution of a number of tridiagonal linear systems every time step. Therefore, it is not clear which scheme is to be favoured regarding the computational time required. Another point is that extension of the computational domain is rather tedious using an ADI technique, whereas for the OEH scheme this extension is straightforward to implement.

3.3. A comparison between the Poisson solvers. The OEH scheme is a fast scheme per time step. Therefore, in order to construct a fast OEH-PC scheme per time step, one needs a fast Poisson solver. In this section we will compare the ICCG method [13], [14] with the MG method MGD5V [9], [19] we employed for the model problem. This comparison is focussed on the computational time required for both methods.

The storage requirements for both methods are approximately the same, and are substantial compared to the storage requirements for the OEH scheme. With respect to the storage requirements, an excellent candidate to combine with the OEH scheme is the MG method MG00 [3]. Unfortunately, at the time of carrying out this research, MG00 was not available in our computer centre, so we decided to compare ICCG with MGD5V.

The ICCG method is an iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, and hence this method can be used for the computation of the pressure. It is an incomplete decomposition method, combined with the conjugate gradient method (cf. [13] and [14]). We used the ICCG $(1,3)$ method from [14]. The MG method MDG5V is a sawtooth multigrid iterative process (i.e., one relaxation-sweep after each coarse grid correction) for the solution of linear second order elliptic boundary value problems, cf. [9] and [19]. This multigrid method uses incomplete line LU-decomposition as relaxation method, a 7-point prolongation and restriction, and a Galerkin approximation for the coarse grid matrices. The ICCG $(1,3)$ process and the MG process were repeated, until the l_2-norm of the residual was less than 10^{-6}.

Using both Poisson solvers, the computations of § 3.1 (for $a=0$) were repeated on a staggered grid with gridsizes $h=k=1/8$, $1/16$, $1/32$ and stepsizes $\tau=h^{-1}$, $\frac{1}{2}h^{-1}, \cdots, 1/1024$. Computations were performed on a Cyber 170-750 computer, and all codes were in standard Fortran 77, except the code for the ICCG method which is written in standard Fortran 66. Parameters of interest in our comparison are: the (CPU-) time (in seconds) needed for the OEH scheme (TOEH), the time needed for the ICCG method (TICCG), the time needed for the MG method (TMG), the ratios $\alpha_1=\text{TICCG}/\text{TOEH}$ and $\alpha_2=\text{TMG}/\text{TOEH}$, and the average number of iteration steps (average over the number of time steps) for either the ICCG method or the MG method (ANIT). In Table 4 we present the results for $h^{-1}=k^{-1}=8, 16, 32$.

From this table, we can draw the following conclusions. For the ICCG method ANIT (and hence α_1) is approximately proportional to $h^{-1}=k^{-1}$, whereas for the MG method ANIT (and hence α_2) is approximately constant. One iteration step of the ICCG method is faster than one iteration step of the MG method, and therefore the ICCG method is faster on coarser grids and the MG method is faster on finer grids. It should be noted that in the ICCG method, the decomposition of the matrix L is

TABLE 4
Comparison between the ICCG *method and the* MG *method.*

ICCG method

	$h^{-1}=k^{-1}=8$				$h^{-1}=k^{-1}=16$				$h^{-1}=k^{-1}=32$			
τ^{-1}	TOEH	TICCG	α_1	ANIT	TOEH	TICCG	α_1	ANIT	TOEH	TICCG	α_1	ANIT
8	0.035	0.090	2.57	7.00								
16	0.070	0.159	2.27	6.06	0.191	1.026	5.37	11.38				
32	0.131	0.313	2.39	5.91	0.410	1.847	4.50	10.09	1.291	14.329	11.10	21.07
64	0.273	0.555	2.03	5.02	0.784	3.401	4.34	9.14	2.543	25.901	10.19	19.17
128	0.563	0.973	1.73	4.27	1.561	6.144	3.94	8.29	5.093	47.019	9.23	17.21
256	1.070	1.582	1.48	3.16	3.100	9.247	2.98	5.96	10.324	84.165	8.22	15.16
512	2.161	2.956	1.37	2.82	6.234	15.991	2.57	4.87	20.558	143.051	6.96	12.68
1024	4.348	4.819	1.11	2.00	12.566	27.302	2.17	3.96	40.657	201.660	4.96	8.59

MG method

	$h^{-1}=k^{-1}=8$				$h^{-1}=k^{-1}=16$				$h^{-1}=k^{-1}=32$			
τ^{-1}	TOEH	TMG	α_2	ANIT	TOEH	TMG	α_2	ANIT	TOEH	TMG	α_2	ANIT
8	0.029	0.207	7.14	5.00								
16	0.071	0.389	5.48	4.69	0.197	0.936	4.75	5.06				
32	0.137	0.669	4.88	4.03	0.384	1.785	4.65	4.78	1.274	5.246	4.12	5.00
64	0.279	1.281	4.59	3.77	0.770	3.044	3.95	4.02	2.546	10.049	3.95	4.78
128	0.543	2.099	3.87	3.01	1.543	5.801	3.76	3.81	5.046	17.075	3.38	4.00
256	1.084	3.161	2.92	2.13	3.143	9.588	3.05	3.00	10.099	32.406	3.21	3.77
512	2.154	6.078	2.82	2.00	6.286	18.024	2.87	2.82	19.874	52.385	2.64	3.01
1024	4.448	12.209	2.74	2.00	12.595	27.252	2.16	2.00	40.210	97.593	2.43	2.71

computed at every time step, whereas in the MG method this is done only once. This will not affect our conclusions seriously, since the computational time required for this decomposition is negligible compared to the computational time needed even for a small number of iterations [13]. Therefore, we may conclude that the MG method is to be preferred to the ICCG method. Also observe that ANIT (and hence the computational time per time step) decreases if we take smaller time steps. The obvious reason for this is that the initial guess of the pressure increment Q^n, for which we use Q^n from the previous time step, improves if we take smaller time steps τ. Finally, although the ICCG method and the MG method are generally considered as fast Poisson solvers, they still require a considerable part of the computational time in the OEH-PC scheme. In our test problem this varies from 53–92 percent for the ICCG method and from 68–88 percent for the MG method (see the columns under α_1 or α_2 in Table 4). This clearly demonstrates that it is very important to use a fast Poisson solver for the construction of a fast OEH-PC scheme.

4. Concluding remarks. In this paper we have constructed the OEH-PC scheme, and demonstrated by two examples that it is a feasible scheme for the computation of incompressible fluid flow. The scheme has a few attractive properties. First, the scheme is very simple as it is (almost) explicit. Therefore, extension to arbitrary domains (also three-dimensional) and to nonuniform grids is straightforward. Also the use of standard upwind differencing renders no problem. Second, the scheme is fast per time step, provided we have a fast Poisson solver for the computation of the pressure. Since the scheme is in fact an explicit scheme, it is easy to vectorize for applications on a supercomputer. Finally, the scheme requires only one array of storage for the computation of the velocity, which is especially advantageous for three-dimensional problems. A drawback of the scheme is the so-called DFF deficiency (see (3.3)), which has in general a negative influence on the accuracy. For many flow problems however, this deficiency will be of only minor importance.

Considered as an ODE time-integration technique, the OEH-PC scheme is second order accurate for the computation of the velocity and first order accurate for the computation of the pressure. Comparing the underlying OEH scheme with the ADI scheme, we note the following. The ADI scheme has in general better stability properties than the OEH scheme. However, the ADI scheme is stepwise more expensive than the OEH scheme, and it would also require more memory. Therefore, we believe that the OEH-PC scheme is a simple alternative to the ADI-PC scheme for many flow problems.

Acknowledgment. The author would like to thank J. van Kan for reading the manuscript.

REFERENCES

[1] T. CEBECI, R. S. HIRSCH, H. B. KELLER AND P. G. WILLIAMS, *Studies of numerical methods for the plane Navier-Stokes equations*, Comput. Meth. Appl. Mech. Engrg., 27 (1981), pp. 13-44.

[2] A. J. CHORIN, *Numerical solution of the Navier-Stokes equations*, Math. Comp., 22 (1968), pp. 745-762.

[3] H. FOERSTER AND K. WITSCH, *Multigrid software for the solution of elliptic problems on rectangular domains*: MG00 (*release* 1), in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, Berlin, 1981, pp. 427-460.

[4] M. FORTIN, R. PEYRET AND R. TEMAM, *Résolution numérique des équations de Navier-Stokes pour un fluide incompressible*, J. Méc., 10 (1971), pp. 357-390.

[5] A. R. GOURLAY, *Hopscotch*: *a fast second-order partial differential equation solver*, J. Inst. Maths. Applics., 6 (1970), pp. 375-390.

[6] A. R. GOURLAY AND G. R. MCGUIRE, *General hopscotch algorithm for the numerical solution of partial differential equations*, J. Inst. Maths. Applics., 7 (1971), pp. 216-227.

[7] P. M. GRESHO, R. L. LEE AND R. L. SANI, *On the time-dependent solution of the incompressible Navier-Stokes equations in two and three dimensions*, in Recent Advances in Numerical Methods in Fluids, Vol. 1, C. Taylor and R. Morgan, eds., Pineridge Press, Swansea, Wales, 1981, pp. 27-79.

[8] F. H. HARLOW AND J. E. WELCH, *Numerical calculation of time-dependent viscous incompressible flow of fluids with free surface*, Phys. Fluids, 8 (1965), pp. 2182-2189.

[9] P. W. HEMKER AND P. M. DE ZEEUW, *Some implementations of multigrid linear system solvers*, in Multigrid Methods For Integral And Differential Equations, The Institute of Mathematics and Its Applications Conference Series, D. J. Paddon and H. Holstein, eds., Oxford University Press, New York, pp. 85-116.

[10] A. C. HINDMARSH, P. M. GRESHO AND D. F. GRIFFITH, *The stability of explicit Euler time-integration for certain finite difference approximations of the multi-dimensional advection-diffusion equation*, Internat. J. Numer. Meth. Fluids, 4 (1984), pp. 853-897.

[11] P. J. VAN DER HOUWEN AND J. G. VERWER, *One-step splitting methods for semi-discrete parabolic equations*, Computing, 22 (1979), pp. 291-309.

[12] J. VAN KAN, *A second-order pressure correction method for viscous incompressible flow*, this Journal, 7 (1986), pp. 870-891.

[13] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148-162.

[14] ———, *Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems*, J. Comput. Phys., 44 (1981), pp. 134-155.

[15] R. PEYRET AND T. D. TAYLOR, *Computational Methods for Fluid Flow*, Springer-Verlag, New York, 1983.

[16] S. M. SCALA AND P. GORDON, *Reflection of a shock wave at a surface*, Phys. Fluids, 9 (1966), pp. 1158-1166.

[17] ———, *Solution of the time-dependent Navier-Stokes equations for the flow around a circular cylinder*, AIAA J., 6 (1968), pp. 815-822.

[18] A. SEGAL, *Aspects of numerical methods for elliptic singular perturbation problems*, this Journal, 3 (1982), pp. 327–349.

[19] P. SONNEVELD, P. WESSELING AND P. M. DE ZEEUW, *Multigrid and conjugate gradient methods as convergence acceleration techniques*, in Multigrid Methods For Integral And Differential Equations, The Institute Of Mathematics And Its Applications Conference Series, D. J. Paddon and H. Holstein, eds., Oxford University Press, New York, 1985, pp. 117–167.

[20] R. TEMAM, *Navier-Stokes Equations*, North-Holland, Amsterdam, 1977.

[21] J. H. M. TEN THIJE BOONKKAMP AND J. G. VERWER, *On the odd-even hopscotch scheme for the numerical integration of time-dependent partial differential equations*, Appl. Num. Math., 3 (1987), pp. 183–193.