Journal of Computational and Applied Mathematics 20 (1987) 393–402
North-Holland

The odd-even hopscotch pressure correction scheme for the incompressible Navier–Stokes equations

J.H.M. TEN THIJE BOONKKAMP
Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Received 29 May 1986

Abstract: The odd-even hopscotch (OEH) scheme, which is a time-integration technique for time-dependent partial differential equations, is applied 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 combined with the pressure correction technique. The resulting scheme is referred to as the odd-even hopscotch pressure correction (OEH-PC) scheme. As a numerical example, we use the OEH-PC scheme to compute the flow through a reservoir. This contribution is based on the work reported in [13]. We refer to that paper for a more comprehensive discussion of the OEH-PC scheme.

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 efficient integration technique (regarding computing time and storage requirements) for the solution of time-dependent partial differential equations (PDEs) [3,4]. We combine the OEH scheme with the pressure correction method, in order to decouple the computation of the velocity and the pressure in a predictor-corrector fashion [1,2,9]. In what follows, the resulting scheme will be referred to as the odd-even hopscotch pressure correction (OEH-PC) scheme.

Consider the incompressible Navier–Stokes equations in conservative form in d space dimensions ($d=2$ or $d=3$) [10]

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

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

where $\boldsymbol{u}$ is the (scaled) velocity, p the (scaled) presure, and Re the Reynolds number. Boundary conditions, to be specified for the velocity field $\boldsymbol{u}$ 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 [8]. 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

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

$$DU = B. \tag{1.4}$$

In (1.3) and (1.4) U, $F(U)$ and P are grid functions defined on a space grid covering Ω, and represent the finite difference approximation to respectively u, $f(u)$ and p. The operators G and D are the finite difference replacements of respectively the gradient- and divergence-operator and B is a term containing boundary values for the velocity u.

We are now ready to define the OEH-PC scheme for the semi-discrete PDE problem (1.3), (1.4). In this section $j = (j_1, \ldots, j_d)$ is a multi-index connected to the grid point x_j of the space grid under consideration and U_j the component of U for x_j (likewise for $F(U)$ and P). 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

$$U_j^{n+1} - \tau\theta_j^{n+1}\big(F(U)_j^{n+1} - (GP)_j^{n+1}\big) = U_j^n + \tau\theta_j^n\big(F(U)_j^n - (GP)_j^n\big). \tag{1.5}$$

Here $\tau = t_{n+1} - t_n$ is the time step, U_j^n stands for the fully discrete approximation to $U_j(t_n)$, and θ is a grid function whose components θ_j^n are defined by [3,4]

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

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.

A somewhat more convenient form of (1.5), for its presentation, reads

$$U^{n+1} = U^n + \tau F_O(U^n) + \tau F_E(U^{n+1}) - \tau(GP^n)_O - \tau(GP^{n+1})_E, \tag{1.7}$$

where F_O is the restriction of F to the odd points (etc.). Note that $F_O + F_E = F$. We shall use this (method of lines [8]) formulation in the remainder of the section. It is also customary to write down two successive steps of (1.7) with step length $\frac{1}{2}\tau$, where the order of explicit and implicit calculations alternate [8,12]

$$\tilde{U} = U^n + \tfrac{1}{2}\tau F_O(U^n) + \tfrac{1}{2}\tau F_E(\tilde{U}) - \tfrac{1}{2}\tau GP^n, \tag{1.8a}$$

$$U^{n+1} = \tilde{U} + \tfrac{1}{2}\tau F_E(\tilde{U}) + \tfrac{1}{2}\tau F_O(U^{n+1}) - \tfrac{1}{2}\tau GP^{n+1}. \tag{1.8b}$$

Alternating between explicit and implicit calculations is the essential feature of the OEH scheme. Scheme (1.8a), (1.8b) is a one-step scheme for the ODE system (1.3) using stepsize τ and $\tilde{U}$ is interpreted as a result from the intermediate time level $n + \frac{1}{2}$. Further we note, that when considered as an ODE solver, this scheme is 2nd order accurate. We also observe that in (1.8a) P is set at time level n and in (1.8b) at level $n + 1$. An alternative, for maintaining 2nd order, is to compute P at time level $n + \frac{1}{2}$ both stages. However, the choice made in (1.8) is better adapted to the pressure correction approach which we shall discuss now.

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

$$DU^{n+1} = B^{n+1}. \tag{1.8c}$$

The computation of U^{n+1} and P^{n+1} requires the simultaneous solution of (1.8b) and (1.8c). In order to avoid this, we follow the known pressure correction approach [1,2,9], 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 $\tilde{\tilde{U}}$. The corrected velocity and pressure (which we hereafter also denote by U^{n+1} and P^{n+1} and hence not should be mixed up with the approximations in (1.8a), (1.8b) and (1.8c)) are then defined by replacing $F_O(U^{n+1})$ in (1.8b) by $F_O(\tilde{\tilde{U}})$:

$$U^{n+1} = \tilde{U} + \tfrac{1}{2}\tau F_E(\tilde{U}) + \tfrac{1}{2}\tau F_O(\tilde{\tilde{U}}) - \tfrac{1}{2}\tau G P^{n+1}, \tag{1.9}$$

together with the discrete continuity equation (1.8c). From (1.9) and the modified equation (1.8b) we trivially get

$$U^{n+1} - \tilde{\tilde{U}} = -\tfrac{1}{2}\tau G Q^n, \quad Q^n + P^{n+1} - P^n. \tag{1.10}$$

The trick of the pressure correction approach is now to multiply (1.10) by D and to write, using (1.8c),

$$LQ^n = \frac{2}{\tau}\left(D\tilde{\tilde{U}} - B^{n+1}\right), \quad L = DG. \tag{1.11}$$

Since $L = DG$ is a discretization of the Laplace operator $\nabla \cdot (\nabla)$, the correction Q^n for the pressure can be obtained by applying a Poisson solver. Once Q^n is known, the new velocity U^{n+1} can be directly determined from (1.10).

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

$$\tilde{U} = U^n + \tfrac{1}{2}\tau F_O(U^n) + \tfrac{1}{2}\tau F_E(\tilde{U}) - \tfrac{1}{2}\tau G P^n, \tag{1.12a}$$

$$\tilde{\tilde{U}} = \tilde{U} + \tfrac{1}{2}\tau F_E(\tilde{U}) + \tfrac{1}{2}\tau F_O(\tilde{\tilde{U}}) - \tfrac{1}{2}\tau G P^n, \tag{1.12b}$$

$$LQ^n = \frac{2}{\tau}\left(D\tilde{\tilde{U}} - B^{n+1}\right), \qquad P^{n+1} = P^n + Q^n, \tag{1.12c}$$

$$U^{n+1} = \tilde{\tilde{U}} - \tfrac{1}{2}\tau G Q^n. \tag{1.12d}$$

We conclude this section with two remarks. First, the 2nd stage (1.12b) can be economized by using its equivalent fast form (Cf. [3,4])

$$\tilde{\tilde{U}}_E = 2\tilde{U}_E - U_E^n, \qquad \tilde{\tilde{U}}_O = \tilde{U}_O + \tfrac{1}{2}\tau F_O(\tilde{\tilde{U}}) - \tfrac{1}{2}\tau (G P^n)_O. \tag{1.12b'}$$

Our implementation is based on this fast form. Second, in the derivation of scheme (1.12) no use has been made of the particular definition of F_O and F_E, except that $F_O + F_E = F$. Consequently, in the spirit of the method of lines formulation [8], pressure correction schemes using other splittings of F, such as ADI, can also be described by (1.12) (see e.g. [9] where an ADI splitting is used).

2. The OEH-PC scheme: space discretization

This section is devoted to the space discretization of the Navier–Stokes problem, which defines the fully discrete OEH-PC scheme. For the sake of presentation, we restrict ourselves to 2-dimensional rectangular domains.

Consider the 2-dimensional incompressible Navier–Stokes equations in conservative form

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

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

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

with boundary conditions

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

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 [6], see Fig. 1. The application of standard, 2nd order central differences on this grid converts (2.1a) and (2.1b) into (cf. (1.3))

$$\dot{U}_{ij} = F_{1,ij}(U, V) - d_x P_{ij}, \qquad i = 1(1)N-1, \quad j = 1(1)M \quad \text{(interior } \times\text{-points)}, \tag{2.3a}$$

$$\dot{V}_{ij} = F_{2,ij}(U, V) - d_y P_{ij}, \qquad i = 1(1)N, \quad j = 1(1)M-1 \quad \text{(interior O-points)}, \tag{2.3b}$$

where

$$F_{1,ij}(U, V) = -\frac{1}{2h}\left(U_{i+1,j}^2 - U_{i-1,j}^2\right) - \frac{1}{2k}\left(U_{i,j+1}\overline{V}_{i,j+1} - U_{i,j-1}\overline{V}_{i,j-1}\right) + \frac{1}{\text{Re}\, h^2}\cdot\left(U_{i+1,j} - 2U_{ij} + U_{i-1,j}\right) + \frac{1}{\text{Re}\, k^2}\left(u_{i,j+1} - 2U_{ij} + U_{i,j-1}\right), \tag{2.4a}$$

$$F_{2,ij}(U, V) = -\frac{1}{2h}\left(\overline{U}_{i+1,j}V_{i+1,j} - \overline{U}_{i-1,j}V_{i-1,j}\right) - \frac{1}{2k}\left(V_{i,j+1}^2 - V_{i,j-1}^2\right) + \frac{1}{\text{Re}\, h^2}\cdot\left(V_{i+1,j} - 2V_{ij} + V_{i-1,j}\right) + \frac{1}{\text{Re}\, k^2}\cdot\left(V_{i,j+1} - 2V_{ij} + V_{i,j-1}\right), \tag{2.4b}$$

$$d_x P_{ij} = \frac{1}{h}\left(P_{i+1,j} - P_{ij}\right), \qquad d_y P_{ij} = \frac{1}{k}\left(P_{i,j+1} - P_{ij}\right). \tag{2.4c,d}$$

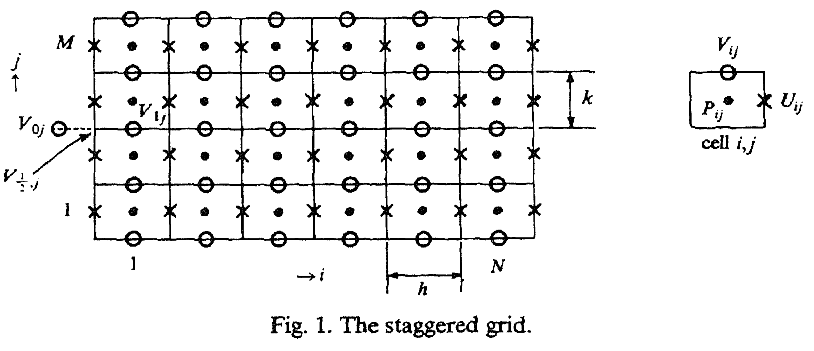

Fig. 1. The staggered grid.

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, respectively, at the corresponding gridpoints. In (2.4a) $\overline{V}_{ij}$ represents an approximation to V in the $\times$-points (points were U is defined); likewise $\overline{U}_{ij}$ represents an approximation to U in the O-points. The values of $\overline{V}_{ij}$ and $\overline{U}_{ij}$ are determined by averaging over neighbouring values of respectively V_{ij} and U_{ij} 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

$$\overline{U}_{ij} = \tfrac{1}{2}\left(U_{ij} + U_{i-1,j+1}\right), \qquad \overline{V}_{ij} = \tfrac{1}{2}\left(V_{ij} + V_{i+1,j-1}\right). \tag{2.5}$$

The space discretization of (2.1), as defined in (2.3), (2.4) determines the vector-function $\boldsymbol{F}(\boldsymbol{U})$ and the operator G in (1.3). Let $\boldsymbol{U} = (U, V)^{\mathrm{T}}$, then $\boldsymbol{F}_{ij}(\boldsymbol{U}) = (F_{1,ij}(U, V), F_{2,ij}(U, V))^{\mathrm{T}}$ and $GP_{ij} = (\mathrm{d}_x P_{ij}, \mathrm{d}_y P_{ij})^{\mathrm{T}}$.

Treatment of the boundary conditions for the velocity is somewhat tedious. Consider e.g. equation (2.4b) in a O-point $(1, j)$, which involves the value V_{0j} outside the computational domain. There are various ways to define the outside value V_{0j}, Cf. [10]. We applied a simple reflection technique, which consists of writing the given velocity $V_{1/2j}$ on Γ as the mean value of the two neighbouring velocities V_{0j} and V_{1j}, so that $V_{0j} = 2V_{1/2j} - V_{1j}$; see Fig. 1.

Equation (2.2) is discretized (using central differences) in all $\cdot$-points as

$$(DU)_{ij} := \frac{1}{h}\left(U_{ij} - U_{i-1,j} + \beta\left(V_{ij} - V_{i,j-1}\right)\right) = 0, \tag{2.6}$$

where $\beta = h/k$. Note that boundary values for U or V occurring in (2.6) are written in the right hand side B (cf. (1.4)). For example, for $j = 1$, equation (2.2) is discretized as

$$(DU)_{i1} = \frac{1}{h}\left(U_{i1} - U_{i-1,1} + \beta V_{i1}\right) = B_{i1} = \frac{1}{k} V_{i0}. \tag{2.6$'$}$$

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

$$\begin{aligned}(LQ)_{ij} = D(GQ)_{ij} &= \frac{1}{h}\left(d_x Q_{ij} - d_x Q_{i-1,j} + \beta\left(d_y Q_{ij} - d_y Q_{i,j-1}\right)\right)\\ &= \frac{1}{h^2}\left(\beta^2 Q_{i,j-1} + Q_{i-1,j} - (2 + 2\beta^2) Q_{ij} + Q_{i+1,j} + \beta^2 Q_{i,j+1}\right),\end{aligned} \tag{2.7}$$

which is the standard 5-point molecule for the Laplace operator. Near a boundary (2.7) takes a different form, because of the different definition of the operator D. For example for $j = 1$, one finds

$$\begin{aligned}(LQ)_{i1} = D(GQ)_{i1} &= \frac{1}{h}\left(d_x Q_{i1} - d_x Q_{i-1,1} + \beta d_y Q_{i1}\right)\\ &= \frac{1}{h^2}\left(Q_{i-1,1} - (2 + \beta^2) Q_{i1} + Q_{i+1,1} + \beta^2 Q_{i2}\right).\end{aligned} \tag{2.7$'$}$$

Now consider equation (1.12c) at the $\cdot$-points $(i, 1)$ $(i = 1(1)N)$. Using (2.6), (2.6′), (2.7) and (2.7′), it is easy to see that $(Q_{i0} - Q_{i1})/k = 2(V_{i0}^{n+1} - V_{i0})/\tau = 0$, which is the (central difference) approximation of $(\partial Q/\partial n)((i - \frac{1}{2})h, 0) = 0$, where n is the outward unit normal on $x = 0$. Hence we see that a Neumann condition for the pressure (-increment) is automatically involved in the scheme.

3. A numerical example: flow through a reservoir

In this section we discuss results of the OEH-PC scheme when used to compute the flow through a reservoir [9] (see Fig. 2). Computations were performed subject to the following initial- and boundary-conditions:

initial conditions: $u = v = 0$ for $t = 0$

boundary conditions:

$$\text{no slip:}\quad u = 0,\quad v = 0,\qquad \text{inlet:}\quad u = 0,\quad v = -432\left(x - \tfrac{1}{4}\right)^2 x\left(1 - e^{-t}\right),$$

$$\text{free slip:}\quad u_y = 0,\quad v = 0;\qquad \text{outlet:}\quad u = 432\left(\tfrac{1}{8} - y\right) y\left(1 - e^{-t}\right),\quad v = 0.$$

Notice that the boundary conditions satisfy

$$\int_{\partial\Omega} \boldsymbol{u} \cdot \boldsymbol{n}\, \mathrm{d}s = \iint_{\Omega} \nabla \cdot \boldsymbol{u}\, \mathrm{d}S = 0,$$

where $\boldsymbol{n}$ 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 a numerical boundary layer at the outlet. This boundary layer may cause oscillations in the solution in the interior domain. Therefore, we have to look for other outlet boundary conditions with minimal influence on the interior flow field. A 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. [5], i.e.

$$\tau_{xx} = -p + \frac{2}{\mathrm{Re}} u_x = 0, \qquad \tau_{xy} = \frac{1}{\mathrm{Re}}\left(u_y + v_x\right) = 0. \tag{3.1}$$

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 hereby 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.

The Poisson solver we used is the multigrid algorithm MGD5V, which is a sawtooth multigrid iterative process (i.e. one relaxation-sweep after each coarse grid correction) for the solution of linear 2nd order elliptic boundary value problems [7,11]. This multigrid method uses incomplete line LU-decomposition as relaxation method, a 7-point prolongation and restriction, and a

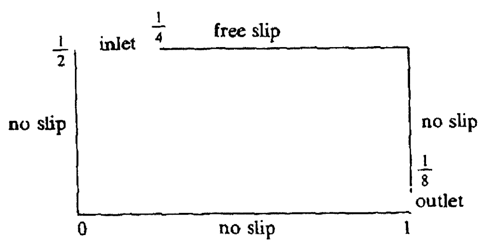

Fig. 2. The reservoir.

Galerkin approximation for the coarse grid matrices. The multigrid process was repeated until the l_2-norm of the residual was less than 10^{-6}.

We have computed the solution of Re = 100(100)800 on the original domain as well as on the extended domain, on a staggered grid with gridsize $h = k = \frac{1}{32}$. Time-integration was performed from $t = 0$ to $t = 4$. The time step τ was bounded by the linearized stability restriction $\tau/h \leqslant \sqrt{2}/(u_{\max})$, where $u_{\max}$ is the (modulus of the) maximum velocity. This (time step) restriction is based on von Neumann analysis applied to the OEH scheme for the corresponding

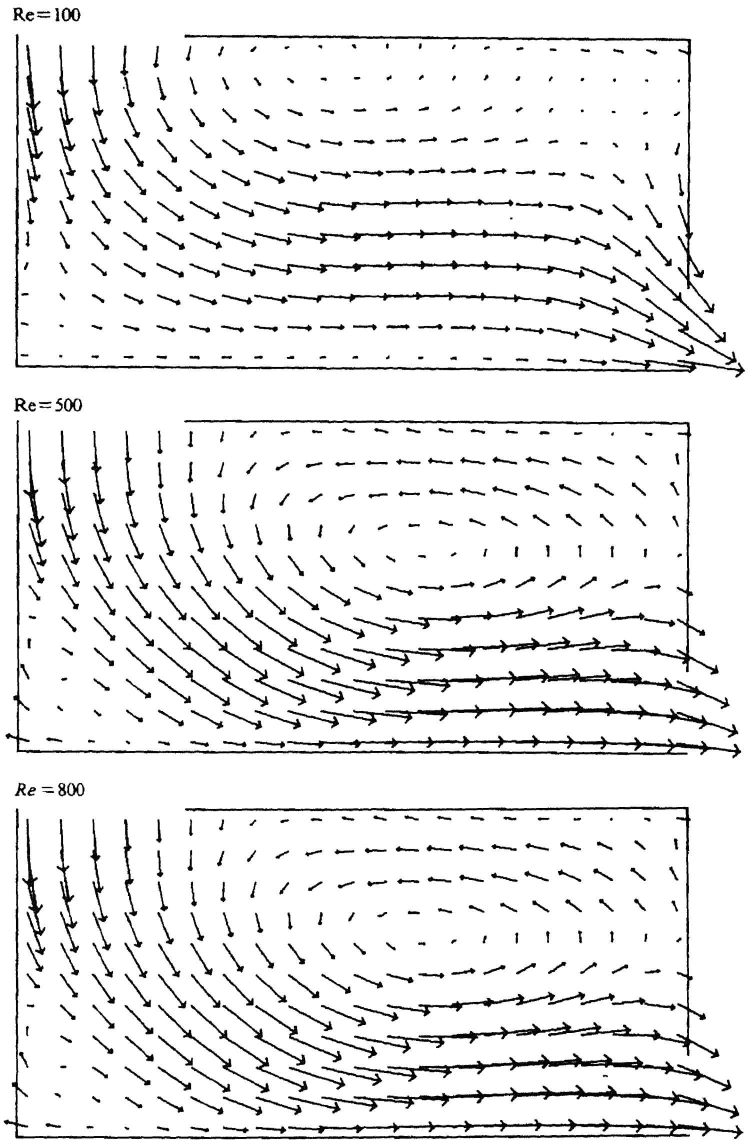

Fig. 3. Velocity field at $t = 4$ for Re = 100, 500 and 800.

linear convection-diffusion equation [12,13]. 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 non-linear 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. In Fig. 3 and 4 you find the velocity and the isobars for respectively Re = 100, 500 and 800 at $t = 4$ computed on the extended domain (the pipe of the extended domain is not shown in these figures).

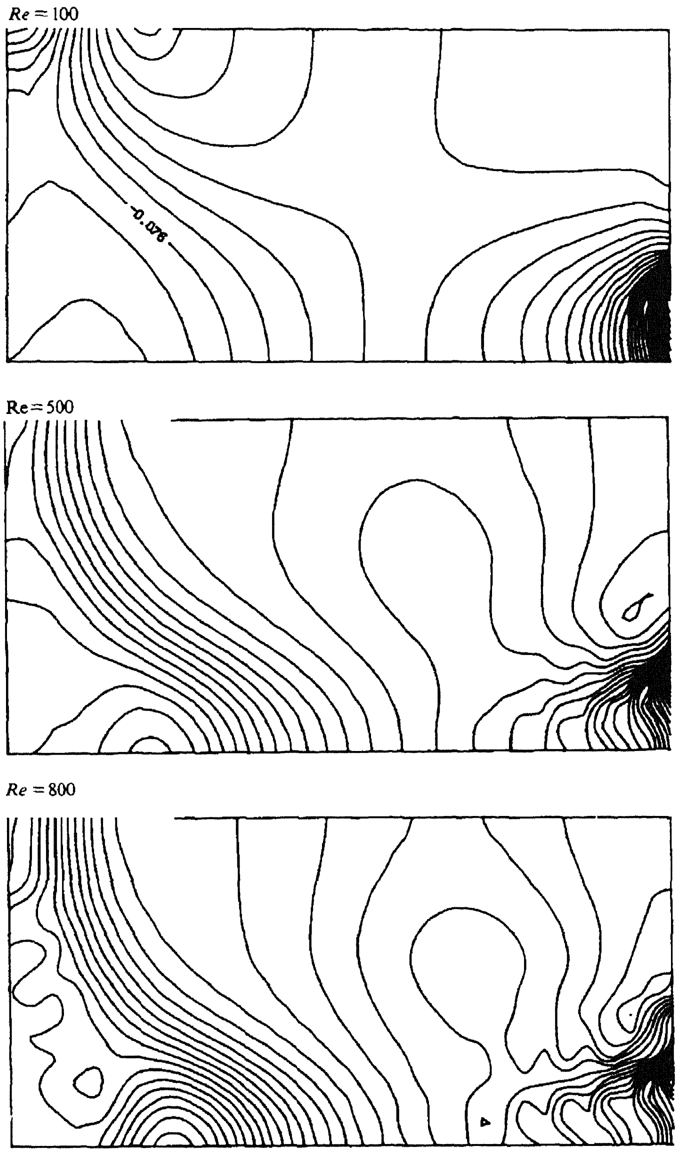

Fig. 4. Isobars at $t = 4$ for Re = 100, 500 and 800.

From our numerical experiments we can draw the following conclusions. For small Re-numbers ($Re \leqslant 200$), 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 virtually free of oscillations. However, small oscillations do occur in the velocity field for $Re > 200$. In this case, the velocity field computed on the extended domain is slightly better (small oscillations) than the velocity field computed on the original domain. The isobars computed on the original domain for $Re > 200$ are not correct, whereas the isobars computed on the extended domain are much more realistic.

We borrowed this model problem from van Kan [9]. 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 the ADI scheme is unconditionally stable, whereas the OEH scheme is only conditionally stable [12,13], so that with respect to stability he 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 [13], and the ADI scheme requires the solution of a number of tridiagonal linear systems. 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 as ADI technique, whereas for the OEH scheme this extension is straightforward to implement. Finally we note that both schemes behave 2nd order in space and time.

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. Eng.* **27** (1981) 13–44.

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

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

[4] A.R. Gourlay and G.R. McGuire, General hopscotch algorithm for the numerical solution of partial differential equations, *J. Inst. Maths. Applics.* **7** (1971) 216–227.

[5] 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: C. Taylor and R. Morgan, Eds., *Recent Advances in Numerical Methods in Fluids, Vol. 1* (Pineridge, 1981) 27–79.

[6] F.H. Harlow and J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluids with free surface, *Phys. Fluids* **8** (1965) 2182–2189.

[7] P.W. Hemker and P.M. de Zeeuw, Some implementations of multigrid linear system solvers, in: D.J. Paddon and H. Holstein, Eds., *Multigrid Methods for Integral and Differential Equations*, The Institute of Mathematics and its Applications Conference Series (Oxford University Press, New York, 1985) 85–116.

[8] P.J. van der Houwen and J.G. Verwer, One-step splitting methods for semi-discrete parabolic equations, *Computing* **22** (1979) 291–309.

[9] J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Stat. Comput.* **7** (1986) 870–891.

[10] R. Peyret and T.D. Taylor, *Computational Methods for Fluid Flow* (Springer, New York, 1983).

[11] P. Sonneveld, P. Wesseling and P.M. de Zeeuw, Multigrid and conjugate gradient methods as convergence acceleration techniques, in: D.J. Paddon and H. Holstein, Eds., *Multigrid Methods for Integral and Differential Equations*, The Institute of Mathematics and its Applications Conference series (Oxford University Press, New York, 1985) 117–167.

[12] 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. Numer. Math.* 3 (1987) 183–193.

[13] J.H.M. ten Thije Boonkkamp, The odd-even hopscotch pressure correction scheme for the incompressible Navier-Stokes equations, report NM-R8615, Centre for Mathematics and Computer Science, Amsterdam, 1986.