JOURNAL OF COMPUTATIONAL PHYSICS 112, 182–209 (1994)

Time Integration Schemes for the Unsteady Incompressible Navier–Stokes Equations

YVES P. MARX

Institut de Machines Hydrauliques et de Mécanique des Fluides, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

Received September 21, 1992; revised March 30, 1993

Several A-stable integration schemes for the solution of the unsteady Navier–Stokes equations for an incompressible fluid are investigated. The numerical procedure follows the method of lines which leads to the solution of a non-linear system of ordinary differential equations after the discretization of the spatial derivatives. The non-linear system is solved iteratively using the artificial compressibility method combined with an implicit upwind scheme. Thorough numerical studies are performed to evaluate the characteristics of the integration schemes. The tests are conducted on two flow problems representative of transient and periodic motions.

1. INTRODUCTION

Due to constraints of computing costs, the development of numerical techniques for fluid flow simulations has focused mainly on steady state problems. However, many applications of interest are inherently unsteady, e.g., turbulent flows in turbo-machinery. The limitation imposed by computer resources is particularly evident in the simulation of unsteady incompressible flows, since at each instant several iterations are necessary to enforce the incompressibility constraint. (Sub-iterations may be necessary for determining at each time step the solution of a pseudo-steady problem, or for solving the pressure Poisson equation that is used for projecting the non-solenoidal velocity field onto the space of divergence-free fields.) With the continuous reduction in computer costs over the past few years, more attention has been devoted to the simulation of unsteady incompressible flows [1–4]. Since the vorticity-stream function formulation is not easily extended to three-dimensional flow problems, the recent solution techniques solve the incompressible formulation of the Navier–Stokes equations using usually the so-called primitive variables. These techniques belong generally to either the segregated methods group [5–8], or to the fractional step methods family [9, 10], or to the artificial compressibility type methods [11, 12]. When a segregated or a fractional step method is employed for simulating incompressible flows, the flow-fields are obtained in two steps. First, a tentative velocity field is evaluated from the solution of the momentum equation. Since this velocity field may not be solenoidal, in a correction step the velocities are projected onto the space of divergence-free fields. The projection is performed using the decomposition of a vector field into solenoidal and irrotational parts,

$$\mathbf{u} = \mathbf{w} + \nabla\phi \qquad \text{with} \quad \nabla \cdot \mathbf{w} = 0.$$

Applying the divergence operator on the decomposition, the irrotational part, and subsequently the incompressible velocity field, can be determined by solving the Poisson equation for the projector ϕ. As the pressure and projector fields are connected [13, 9], the projection step provides also the pressure field. With these procedures, only semi-implicit methods can be employed since, by construction, the momentum and continuity equations are solved separately. Therefore, the admissible time step is restricted by stability conditions. As a consequence, the efficiency of these procedures depends strongly on the Poisson solver employed in the projection step. As opposed to the previous methods, which can be applied directly to the unsteady equations, the artificial compressibility method [11] was initially designed for solving the steady equations. The procedure consists of preconditioning the system by artificially adding in the continuity equation a pseudo-time derivative of the pressure which is multiplied by a positive constant. For the solution of unsteady problems, the procedure was initially extended by taking small values for the positive constant [14]. This procedure, however, produces very stiff systems that are difficult to solve numerically. A better extension of the artificial compressibility method is to use the method for solving iteratively the system that arises from an implicit approximation of the unsteady Navier–Stokes equations [15]. The efficiency of the procedure then depends on the work necessary for solving the

0021-9991/94 $6.00

pseudo-steady equations at each time step and on the value of the time step that can be used.

In the present work, the unsteady Navier–Stokes equations are discretized in time with a fully implicit scheme, Section 2. Thus, at each time step a non-linear system resembling the steady form of the Navier–Stokes equations has to be solved. The artificial compressibility method coupled with a classical ADI-type implicit upwind scheme, Section 3, is used for that purpose. The iterative scheme serves then in all the computations for evaluating the efficiency of several A-stable time integration schemes, Section 4. This evaluation is conducted on two flow problems characterizing transient and periodic unsteadiness. For each integration scheme, several strategies for optimizing the simulations has been investigated. Specifically, the influence of the time step on the accuracy of the solution, and on the number of iterations necessary to solve at each time step the pseudo-steady problems, has been studied. The effect of the convergence level used in the solution of the pseudo-steady problems has also been examined. The investigations show that even if the same basic procedure is employed for simulating unsteady flows, the efficiency of the simulation depends significantly on the integration scheme. Analyzing the numerical experiments, guidelines for designing the integration schemes are formulated, Section 5. An implicit Runge–Kutta scheme satisfying all the desired criteria is proposed. Finally, from the tests conducted it is also shown that if an appropriate correction is performed at each time step, accurate simulations can be obtained without solving exactly the pseudo-steady problems.

2. THE UNSTEADY PROCEDURE AND TIME DISCRETIZATIONS

For the onset work, the method of line is employed. It consists in separating the time and spatial discretizations. The unsteady Navier–Stokes and continuity equations are written as

$$\tilde{I}\frac{\partial U}{\partial t} + \mathcal{N}(U) = 0, \tag{1}$$

where

$$\tilde{I} = \mathrm{diag}(0, 1, 1, 1); \qquad U = [p, u, v, w]^{\mathrm{T}},$$

and $\mathcal{N}$ represents the steady incompressible Navier–Stokes operator.

The first step consists in approximating the spatial operator. Equation (1) then nearly becomes a system of almost ordinary differential equations

$$\left(\tilde{I}\frac{dU}{dt}\right)_{i,j,k} + \mathcal{N}_h(U)_{i,j,k} = 0, \tag{2}$$

for the vector $[U_{i,j,k}]$, where $U_{i,j,k}$ includes the cell-averaged values of the pressure and velocities at the cell (i, j, k). In (2), $\mathcal{N}_h$ represents the discretized form of the steady Navier–Stokes equations. System (2) is not completely a system of ordinary differential equations because the incompressible continuity equation has no time derivative term and therefore cannot be transformed into an ODE. However, despite this peculiarity, most methods designed for solving ODEs can be applied to the solution of (1).

2.1. Discretization Schemes

To ensure that the velocity field remains divergence-free at each discretized time step, the continuity equation must be enforced at the new time level. For the momentum equation, a wide range of time discretization schemes can be chosen. Desirable properties for those time stepping schemes are [16],

- strong A-stability,
- non-dissipativity.

In the present work, four widely-used A-stable schemes have been investigated; the implicit Euler scheme, the second-order backward-differencing scheme, the Crank–Nicolson scheme, and diagonal implicit Runge–Kutta schemes [17]. Runge–Kutta schemes are defined by several parameters. This versatility can be used to design schemes having the desired properties. Since the solution of pseudo-steady problems necessitates more iterations per time step with multi-stage schemes than with one-stage schemes, only two-stage schemes were considered. Two Runge–Kutta (DIRK 2) schemes of the form

$$\begin{aligned} U^{(1)} &= U^n - \Delta t\,(\beta_{10} N(U^n) + \beta_{11} N(U^{(1)})), \\ U^{n+1} &= U^n - \Delta t(\beta_{20} N(U^n) + \beta_{21} N(U^{(1)}) + \beta_{22} N(U^{n+1})) \end{aligned} \tag{3}$$

were designed. These two schemes are second-order accurate, strongly A-stable, and non-dissipative. For the first DIRK 2 scheme, accuracy was maximized by minimizing the amplitude and the phase error of a periodic signal. The second DIRK 2 scheme was designed with the objective of improving the smoothing properties of the first DIRK 2 scheme without reducing significantly its accuracy. The β coefficients resulting from the above optimization are

DIRK 2.1. Scheme $\beta_{10} = 0$, $\beta_{11} = 0.2060$, $\beta_{20} = -0.2698$, $\beta_{21} = 0.9698$, $\beta_{22} = 0.3$.

DIRK 2.2. Scheme $\beta_{10} = 0$, $\beta_{11} = 0.2651$, $\beta_{20} = -0.0545$, $\beta_{21} = 0.7545$, $\beta_{22} = 0.3$.

With either of the integration schemes described above, the discretized form of the unsteady Navier–Stokes equations can be written as

$$\alpha \tilde{I} U^{n+1}_{i,j,k} + \mathcal{N}_h(U^{n+1})_{i,j,k} = \Phi^n_{i,j,k}, \tag{4}$$

where

Φ^n is a (known) source term,

α is a coefficient.

System (4) is non-linear and resembles the steady state Navier–Stokes equations. The procedure employed to solve this pseudo-steady system is detailed in the next section. Before that, the consequence of an approximate solution of (4) is discussed.

2.2. Time Correction

Generally (4) is solved approximately, and only a tentative solution is computed. For such a solution

$$\alpha\tilde{I}\bar{U}^{n+1}_{i,j,k} + \mathcal{N}_h(\bar{U}^{n+1})_{i,j,k} = \Phi^n_{i,j,k} + \varepsilon_{i,j,k},$$

where ε is some (small) tolerance parameter and the $(^-)$ denotes an approximate solution.

It is simplest to take $U^{n+1} = \bar{U}^{n+1}$. In this case, in addition to the truncation error a "resolution" error is introduced at each iteration. This error can be easily suppressed if a fixed point integration is performed after the computation of $\bar{U}^{n+1}$. The new field U^{n+1} is then obtained from

$$\alpha\tilde{I}\tilde{U}^{n+1}_{i,j,k} + \mathcal{N}_h(\bar{U}^{n+1})_{i,j,k} = \Phi^n_{i,j,k}. \tag{5}$$

However, explicit integration could become unstable for large Δt. A third possibility is to calculate $\bar{U}^{n+1}$ and $\tilde{U}^{n+1}$ and to employ them both in the determination of the source term at the new time level. In the computation of Φ^{n+1}, $\bar{U}^{n+1}$ is employed to calculate terms like $\mathcal{N}_h(\bar{U}^{n+1})$ and $\tilde{U}^{n+1}$ is employed to calculate terms like $\sigma\tilde{U}^{n+1}/\Delta t$, where σ is a fixed coefficient. With this third procedure, $\bar{U}^{n+1}$ is also used as the initial guess for U^{n+2}. The difference between the three techniques can be illustrated on the Crank–Nicolson scheme by considering the source term at the $(n+2)$ time level. With the first procedure, Φ^{n+1} is

$$\Phi^{n+1} = 2\tilde{I}\frac{\bar{U}^{n+1}}{\Delta t} - \mathcal{N}_h(\bar{U}^{n+1}).$$

With the second procedure it is

$$\Phi^{n+1} = 2\tilde{I}\frac{\tilde{U}^{n+1}}{\Delta t} - \mathcal{N}_h(\tilde{U}^{n+1}),$$

and, finally, with the third procedure it is

$$\Phi^{n+1} = 2\tilde{I}\frac{\tilde{U}^{n+1}}{\Delta t} - \mathcal{N}_h(\bar{U}^{n+1}).$$

Replacing $2\tilde{I}\bar{U}^{n+1}/\Delta t$ or $2\tilde{I}\tilde{U}^{n+1}/\Delta t$ by its expression, the system solved at time $(n+2)\,\Delta t$ is then

$$2\tilde{I}\frac{U^{n+2}}{\Delta t} + \mathcal{N}_h(U^{n+2}) = 2\tilde{I}\frac{U^n}{\Delta t} - \mathcal{N}_h(U^n) - 2\mathcal{N}_h(\bar{U}^{n+1}) + \varepsilon^{n+1} \quad \textbf{(no correction)},$$

$$2\tilde{I}\frac{U^{n+2}}{\Delta t} + \mathcal{N}_h(U^{n+2}) = 2\tilde{I}\frac{U^n}{\Delta t} - \mathcal{N}_h(U^n) - 2\mathcal{N}_h(\tilde{U}^{n+1}) \quad \textbf{(correction (O))},$$

$$2\tilde{I}\frac{U^{n+2}}{\Delta t} + \mathcal{N}_h(U^{n+2}) = 2\tilde{I}\frac{U^n}{\Delta t} - \mathcal{N}_h(U^n) - 2\mathcal{N}_h(\bar{U}^{n+1}) \quad \textbf{(correction (N))}.$$

Comparing these expressions with the exact expression for the Crank–Nicolson scheme

$$2\tilde{I}\frac{U^{n+2}}{\Delta t} + \mathcal{N}_h(U^{n+2}) = 2\tilde{I}\frac{U^n}{\Delta t} - \mathcal{N}_h(U^n) - 2\mathcal{N}_h(U^{n+1}),$$

it clearly appears that the three procedures described above correspond to different approximations of the Crank–Nicolson scheme. It can be observed that, whereas with the first procedure, named "no correction," two types of errors, ε^{n+1} and $2(\mathcal{N}_h(U^{n+1}) - \mathcal{N}_h(\bar{U}^{n+1}))$ are introduced, with the second and third procedures, named "correction (O)" and "correction (N)", the resolution error ε^{n+1} has been removed. The "correction (N)" procedure is equivalent to the "General Corrective Procedure" of Hirt and Harlow [18].

3. SOLUTION OF THE PSEUDO-STEADY EQUATIONS

As explained in the previous section, any technique developed for solving the steady incompressible equations can be employed to solve the discretized form of the unsteady Navier–Stokes equations (4). In the present work, the artificial compressibility method is used.

3.1. Spatial Discretization

The system (4) is considered as being composed of four parts; the source terms $\Phi^n_{i,j,k}$, the unsteady term $\alpha\tilde{I}U^{n+1}_{i,j,k}$, the advection, and the diffusion terms. Each term is discretized separately.

Since a finite volume method is employed, the discretization of the unsteady term is simply

$$(\mathscr{V}\alpha\tilde{I}U^{n+1})_{i,j,k},$$

where $\mathscr{V}_{i,j,k}$ represents the volume of the i, j, k cell.

The diffusion terms are approximated using a centered discretization. The gradients at each cell face are evaluated from an application of the Gauss theorem on co-volumes.

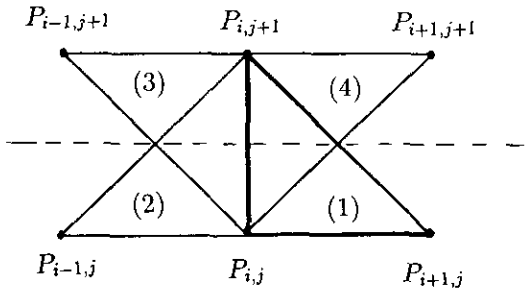

FIGURE 1

Triangular or tetrahedra co-volumes are used in respectively two and three dimensions. For instance, in two dimensions the gradient at the cell face $i, j+\frac{1}{2}$ are obtained by averaging the gradients calculated on the four triangles displayed in Fig. 1. Thus, the gradient at the $i, j+\frac{1}{2}$ cell face is

$$\nabla u_{i,j+1/2} = \tfrac{1}{4}(\nabla u_{P_{i,j}P_{i+1,j}P_{i,j+1}} + \nabla u_{P_{i,j}P_{i-1,j}P_{i,j+1}} + \nabla u_{P_{i,j}P_{i-1,j+1}P_{i,j+1}} + \nabla u_{P_{i,j}P_{i+1,j+1}P_{i,j+1}})$$

with

$$\nabla u_{P_{i,j}P_{i\pm1,j}P_{i,j\pm1}} = \frac{1}{(\mathbf{l}_i^* \times \mathbf{l}_j^*)\cdot \mathbf{k}} [\pm(u_{i\pm1,j} - u_{i,j})\,\mathbf{n}_i^* \pm (u_{i,j\pm1} - u_{i,j})\,\mathbf{n}_j^*].$$

The vectors $\mathbf{l}_i^*$ and $\mathbf{l}_j^*$ represent the tangential vectors and the vectors $\mathbf{n}_i^*$ and $\mathbf{n}_j^*$ are the normal vectors. These vectors are defined by

$$\mathbf{l}_i^* = P_{i,j}P_{i\pm1,j}, \qquad \mathbf{l}_j^* = P_{i,j}P_{i,j\pm1}.$$
$$\mathbf{n}_i^* = \mathbf{l}_j^* \times \mathbf{k}, \qquad \mathbf{n}_j^* = \mathbf{k} \times \mathbf{l}_i^*.$$

The advection terms are evaluated using an upwind approximation derived from a classical compressible discretization [19]. Since the system formed by the advection terms is hyperbolic when the artificial compressibility method is employed, compressible techniques can easily be extended to solve incompressible flow equations. The discretization of the advection terms is

$$\sum_{\mathscr{C}_{i,j,k}} (\mathscr{P}^{-1}F\cdot\mathbf{n})\,\mathscr{S},$$

where $\mathscr{C}_{i,j,k}$ is the set of the faces of the cell i, j, k, $\mathbf{n}$ is the outward normal, $\mathscr{S}$ is a cell surface, F is the inviscid flux at the cell face, and $\mathscr{P} = \mathrm{diag}(1/c^2, 1, 1, 1)$. The matrix $\mathscr{P}$ is the artificial compressibility preconditioning matrix. Its role is to reduce the stiffness of system (4). This can be achieved by adjusting the parameter c^2. Theoretical analysis [20] as well as numerical experiments [21] have shown that the artificial compressibility coefficient c^2 that gives near optimal convergence rates can be estimated from

$$c^2 = 3 \max_{i,j,k} (u_{i,j,k}^2 + v_{i,j,k}^2 + w_{i,j,k}^2). \tag{6}$$

Defining the Jacobian matrix at the cell face $i+\frac{1}{2}$ by

$$A(\hat{U}) = \frac{\partial \mathscr{P}^{-1}F}{\partial U}\left(\frac{U_{Ri+1/2} + U_{Li+1/2}}{2}\right),$$

the inviscid flux at the cell face calculated using Roe's scheme [22] is

$$\mathscr{P}^{-1}F_{i+1/2} = \tfrac{1}{2}(\mathscr{P}^{-1}F(U_{Li+1/2}) + \mathscr{P}^{-1}F(U_{Ri+1/2})) - \tfrac{1}{2}\sum_{k=1}^{n} r_k\,|\lambda_k|\,l_k(U_{Ri+1/2} - U_{Li+1/2}), \tag{7}$$

where λ_k are the eigenvalues of $A(\hat{U})$ and r_k and l_k are respectively the right and left eigenvectors of $A(\hat{U})$.

The values $U_{Li+1/2}$, $U_{Ri+1/2}$ are reconstructed from the cell averaged values U_{i+l} using

$$U_{Li+1/2} = U_i + \frac{h_i}{6}[4\delta U_{i+1/2} + 2\delta U_{i-1/2}],$$

$$U_{Ri+1/2} = U_{i+1} - \frac{h_{i+1}}{6}[4\delta U_{i+1/2} + 2\delta U_{i+3/2}],$$

with

$$\delta U_{i+1/2} = \frac{U_{i+1} - U_i}{h_{i+1} + h_i},$$

and h_i the distance between the faces $i+\frac{1}{2}$ and $i-\frac{1}{2}$.

Since the source term is a combination of the unsteady term, the advection and the diffusion terms at time $n\,\Delta t$, its discretization is fully defined by the above discretizations.

3.2. Iterative Procedure

The flow-field U^{n+1} is obtained using the iterative procedure

$$U^{n+1,\nu+1} = U^{n+1,\nu} + \delta U,$$
$$\mathscr{L}(U^{n+1,\nu})\,\delta U = \Delta U. \tag{8}$$

The quantity ΔU *is proportional to the residual of the* pseudo-steady equation,

$$\frac{\Delta U}{\Delta \tau} i, j, k + \alpha \tilde{I} U_{i,j,k}^{n+1,\nu} + \mathscr{N}_h(U_{i,j,k}^{n+1,\nu}) = \Phi_{i,j,k}^{n}, \tag{9}$$

where $\Delta\tau$ is a pseudo-time step.

The function of operator $\mathscr{L}$ is to convect and diffuse throughout the computational domain the locally calculated residuals. The construction of the operator $\mathscr{L}$ is detailed below for the two-dimensional Cartesian case only, but the same sequence can also serve to derive the three-dimensional operator for structured non-Cartesian meshes. When $\mathscr{L}$ is the exact linearization of ΔU, $\mathscr{L} = \partial(\Delta U)/\partial U$, the iterative procedure is a Newton method. Despite its quadratic convergence, the Newton method may not be the optimal method for calculating the solution of the system, since the calculation of the exact Jacobian matrix $\partial(\Delta U)/\partial U$ may be difficult and since an expensive direct solver must be usually employed to invert the implicit system. Furthermore, the quadratic convergence is only achieved for flow-fields that are sufficiently close to the solution. Therefore, only an approximation of $\partial(\Delta U)/\partial U$ has been used. This approximation was derived in several steps. The derivation begins with a first-order upwind approximation of the quasi-linear form of (9),

$$\begin{aligned}\mathscr{L}_1(U) = \Big\{ & I + \alpha\,\Delta\tau\,\tilde{I} + \Delta\tau \\ & \times \Big[A^+_{i-1/2,j} \frac{\Delta^-_x}{\Delta x} i,j + A^-_{i+1/2,j} \frac{\Delta^+_x}{\Delta x} i,j \\ & + B^+_{i,j-1/2} \frac{\Delta^-_y}{\Delta y} i,j + B^-_{i,j+1/2} \frac{\Delta^+_y}{\Delta y} i,j \\ & - \mu\tilde{I} \frac{\Delta^2_x}{\Delta x^2} i,j - \mu\tilde{I} \frac{\Delta^2_y}{\Delta y^2} i,j \Big] \Big\}\end{aligned}$$

with

$$\begin{aligned}(\Delta^{\pm}_x U)_{i,j} &= \pm(U_{i\pm1,j} - U_{i,j}), \\ (\Delta^{\pm}_y U)_{i,j} &= \pm(U_{i,j\pm1} - U_{i,j}), \\ (\Delta^2_x U)_{i,j} &= U_{i+1,j} - 2U_{i,j} + U_{i-1,j}, \\ (\Delta^2_y U)_{i,j} &= U_{i,j+1} - 2U_{i,j} + U_{i,j-1}.\end{aligned}$$

The matrices $A^{\pm}$ and $B^{\pm}$ are respectively the positive and negative part of the Jacobian matrices of the inviscid flux in the x and y directions,

$$A^{\pm} = R_x \frac{\Lambda_x \pm |\Lambda_x|}{2} R_x^{-1}, \qquad B^{\pm} = R_y \frac{\Lambda_y \pm |\Lambda_y|}{2} R_y^{-1}.$$

The three-dimensional form of the matrices $A^{\pm}_{\eta}$ that corresponds to the positive and negative part of the Jacobian matrices of the inviscid flux in the direction of the vector $\mathbf{n}$ is given in Appendix A.

An implicit procedure using $\mathscr{L} = \mathscr{L}_1$ was employed in [2]. This procedure necessitates the use of relaxation schemes to invert the implicit system. For two-dimensional flow problems, the implicit system (8) can be inverted efficiently with a slightly modified form of the multigrid method described in [23]. However, the robustness and the efficiency of this multigrid method results from an appropriate combination of zebra relaxations with semi-coarsening. To retain the same robustness and efficiency in three-dimensions, complicated plane relaxations must be used in place of the line relaxations. Therefore, it was preferred in the present work to use for the implicit operator $\mathscr{L}$ the factorized operator $\mathscr{L}_2$,

$$\begin{aligned}\mathscr{L}_2(U) = (I + \alpha\,\Delta\tau\,\tilde{I}) \times \Big\{ & I + (I + \alpha\,\Delta\tau\,\tilde{I})^{-1}\,\Delta\tau \\ & \times \Big[A^+_{i-1/2,j} \frac{\Delta^-_x}{\Delta x} i,j + A^-_{i+1/2,j} \frac{\Delta^+_x}{\Delta x} i,j - \mu\tilde{I} \frac{\Delta^2_x}{\Delta x^2} i,j \Big] \Big\} \\ & \times \Big\{ I + (I + \alpha\,\Delta\tau\,\tilde{I})^{-1}\,\Delta\tau \\ & \times \Big[B^+_{i,j-1/2} \frac{\Delta^-_y}{\Delta y} i,j + B^-_{i,j+1/2} \frac{\Delta^+_y}{\Delta y} i,j - \mu\tilde{I} \frac{\Delta^2_y}{\Delta y^2} i,j \Big] \Big\}.\end{aligned}$$

With this operator, the solution of the implicit system (8) is not more difficult in three dimensions than in two dimensions. In both cases, systems like

$$\begin{aligned}\mathscr{L}_{x2}(U)\,\delta U = & \Big\{ I + \alpha\,\Delta\tau\,\tilde{I} + \Delta\tau \Big[A^+_{i-1/2,j} \frac{\Delta^-_x}{\Delta x} i,j \\ & + A^-_{i+1/2,j} \frac{\Delta^+_x}{\Delta x} i,j - \mu\tilde{I} \frac{\Delta^2_x}{\Delta x^2} i,j \Big] \Big\} \delta U \\ & = (I + \alpha\,\Delta\tau\,\tilde{I})\,\Delta U \end{aligned} \tag{10}$$

have to be solved. This is achieved easily, since only block-tridiagonal matrices must be inverted. In the steady state case, $\alpha = 0$, the operator $\mathscr{L}_{x2}$ can be further simplified using the decomposition of the Jacobian matrices $A^{\pm}$ and replacing $\mu\tilde{I}$ by μI [24]. The latter simplification is justified for μ sufficiently small. With these simplifications, the system (10) can be written as

$$\begin{aligned}\mathscr{L}_{x3}(U)\,\delta U = R_x \Big\{ & I + \Delta\tau \Big[A^+_{i-1/2,j} \frac{\Delta^-_x}{\Delta x} i,j \\ & + A^-_{i+1/2,j} \frac{\Delta^+_x}{\Delta x} i,j - \mu \frac{\Delta^2_x}{\Delta x^2} i,j \Big] \Big\} R_x^{-1}\,\delta U \\ & = (I + \alpha\,\Delta\tau\,\tilde{I})\,\Delta U. \end{aligned} \tag{11}$$

System (11) is the "diagonal" form of (10). The "diagonal" form (11) is simpler than the original form (10) since only *scalar* tridiagonal matrices must be inverted. When α is not zero, as in the unsteady case, the "diagonal" form can also

be obtained. The term $\alpha\,\Delta\tau\,\tilde{I}$ is replaced by $R_x(\alpha\,\Delta\tau\,I)\,R_x^{-1}$. Then, a system similar to

$$\mathscr{L}_{x3}(U)\,\delta U = R_x\left\{I+\alpha\,\Delta\tau\,I+\Delta\tau\left[A^+_{i-1/2,j}\frac{\Delta_x^-}{\Delta x}i,j + A^-_{i+1/2,j}\frac{\Delta_x^+}{\Delta x}i,j-\mu\frac{\Delta_x^2}{\Delta x^2}i,j\right]\right\}R_x^{-1}\,\delta U = (I+\alpha\,\Delta\tau\,\tilde{I})\,\Delta U \tag{12}$$

has to be solved in each direction. The simplification will not alter significantly the system only if $\alpha\,\Delta\tau$ is small. Since $\alpha=\sigma/\Delta t$, where σ is a coefficient that depends on the time stepping scheme employed, the "diagonal" form (12) is legitimate only if the artificial time step $\Delta\tau$ is much smaller than the real time step Δt.

When $\Delta t \ll \Delta\tau$, or more precisely when $\alpha\,\Delta\tau$ is large compared to the pseudo-CFL number, the system (10) can also be simplified in

$$\mathscr{L}_{x4}(U)\,\delta U = \left\{I+\alpha\,\Delta\tau\,\tilde{I}+\Delta\tau\left[\tilde{A}^+_{i-1/2,j}\frac{\Delta_x^-}{\Delta x}i,j + \tilde{A}^-_{i+1/2,j}\frac{\Delta_x^+}{\Delta x}i,j-\mu\tilde{I}\frac{\Delta_x^2}{\Delta x^2}i,j\right]\right\}\delta U = (I+\alpha\,\Delta\tau\,\tilde{I})\,\Delta U. \tag{13}$$

The matrices $\tilde{A}^{\pm}$ are defined as follows: For the line corresponding to the continuity equation, the matrices $\tilde{A}^{\pm}$ and $A^{\pm}$ have identical elements. For the other lines, $\tilde{A}^{\pm}$ is composed of only zero elements. With the form (13), the velocity corrections are immediately obtained. They then serve to calculate the pressure correction. Equation (13) is thus even simpler than Eq. (12) since only *one* scalar tridiagonal matrix has to be inverted.

If Δt is not large or small compared to $\Delta\tau$, the diagonalization procedures (12) and (13) alter the implicit operator $\mathscr{L}_2$ too much and are not recommended. In this (general) case, the non-diagonal form (10) must be used. A "paradox" is thus found. Since the unsteady term enforces the diagonal of the system (4), the solution of the unsteady system (Eq. (4) with $\alpha\neq 0$) should be easier than the solution of the steady system ($\alpha=0$). However, it found that when an ADI-type operator is used, the unsteady term prevents usually the "diagonalization" of the implicit system. Therefore, more computing time is necessary to perform one pseudo-steady iteration on the unsteady system than on the steady system.

In summary, the iterative procedure employed to solve the pseudo-steady equation (4) is done as follows. Starting from $U^{n+1,0}=U^n$, successive approximations of U^{n+1} are calculated using Eqs. (8). One iteration is composed of two stages. First, an explicit high-order approximation ΔU of $U^{n+1,\nu+1}-U^{n+1,\nu}$ is computed from the solution of Eq. (9). This tentative estimate of the increment is then corrected using the low-order operator $\mathscr{L}$. The implicit correction stage is employed to improve the stability and to increase the convergence rate of the scheme. It has no effect on the accuracy of the scheme because at convergence $\delta U=\Delta U=0$, in which case U^{n+1} depends only on Eq. (9).

4. NUMERICAL INVESTIGATIONS

4.1. Impulsively Started Cylinder, Re = 3000

The procedure for the unsteady solution described in Section 2 was first validated. The test case chosen is the flow-field generated by an impulsively started cylinder at $\mathrm{Re}=UD/\mu=3000$, where U is the velocity of the cylinder and D is its diameter. This flow was chosen because during its initial (symmetric) phase, a complex flow pattern forms behind the cylinder. The flow generated is also well documented; the present simulation can thus be compared with experiments [25] as well as with other calculations [26]. The present simulation was performed on a half cylinder using the most accurate time integration scheme considered in Section 2, i.e., DIRK 2.1, with a non-dimensional time step $\Delta t=\Delta t^*(U/D)=0.2$. A 200×120 cylindrical mesh with clustering of the points near the cylinder's surface was employed. The outer boundary was located at $R_\infty=10D$. The initial solution was computed by integrating the unsteady equations from a uniform free-stream flow using the Euler scheme with a very small time step. This first step serves to impose an initial flow-field that satisfies the boundary conditions and the continuity constraint.

The evolution of the primary separation length obtained from the numerical simulation is given in Fig. 2. From Fig. 2, it appears that the present computation agrees very well with both the experiment and the other computation. (The results in [26] were obtained through the solution of the stream-function/vorticity form of the Navier–Stokes equations on a 161×101 mesh.) It can be observed from Fig. 3 that not only the separation length is in good agree-

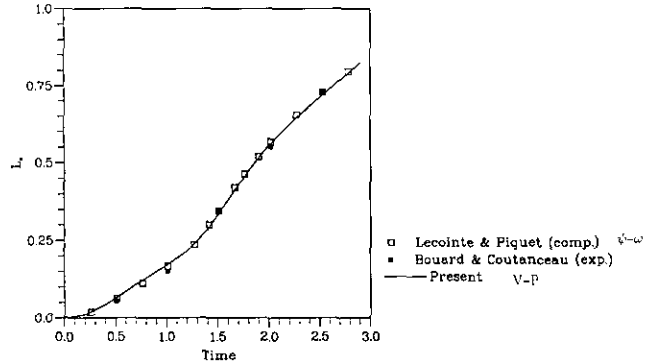

FIGURE 2

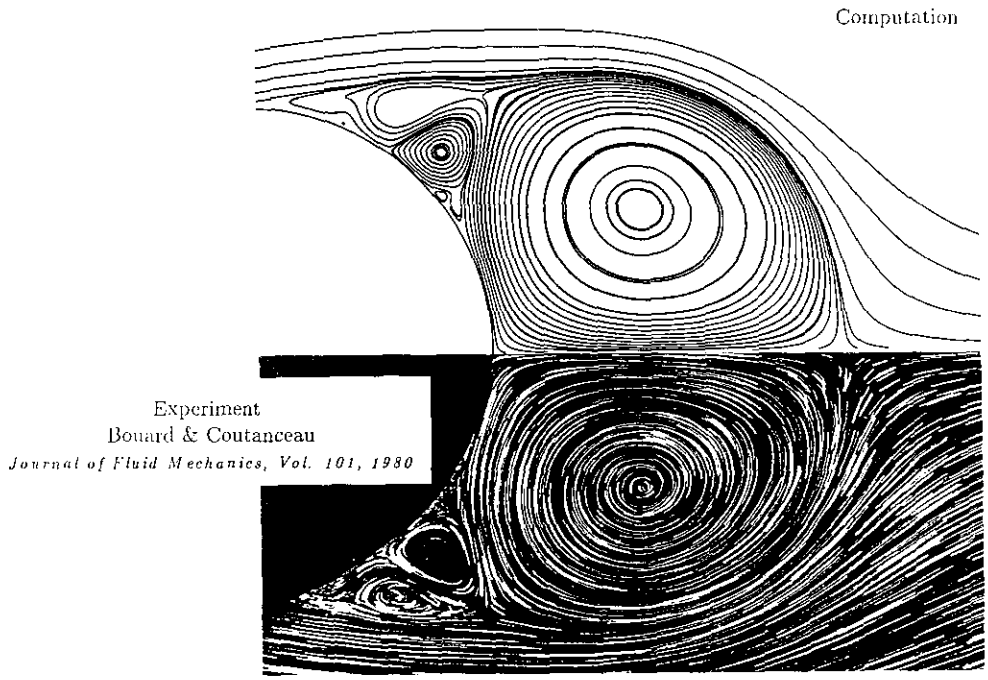

FIGURE 3

ment with the experiment, but the main features of the flow-field are also well captured in the present computation. However, it should be pointed out that the quantities compared in Fig. 3 are not exactly equivalent: in the experiment the particle traces are shown, whereas in the computation the lines correspond to the instantaneous streamlines.

4.2. 2D Driven Cavity Flow, Re = 1000

The lid-driven cavity problem for Re = 1000 was used to evaluate the influence of the implicit operator $\mathscr{L}$ on the convergence of the procedure. Since an accurate representation of the transient phase was of no concern in this series of calculations, all the simulations were done using the first-order Euler scheme on a uniform mesh of 33×33 grid points. The systematic variation of the true time step shows that

— when $\alpha\, \Delta\tau \leqslant \mathrm{CFL}/10$, the "diagonal" form $\mathscr{L}_3$ of the implicit operator and the non-diagonal form $\mathscr{L}_2$ require nearly the same number of iterations to converge at each time step of the pseudo-steady system (4). Since the inversion of the "diagonal" form is 20% less expensive than the inversion of the non-diagonal form, when $\alpha\, \Delta\tau \leqslant \mathrm{CFL}/10$ the use of the "diagonal" form $\mathscr{L}_3$ saves CPU time by approximately 10% to 20%.

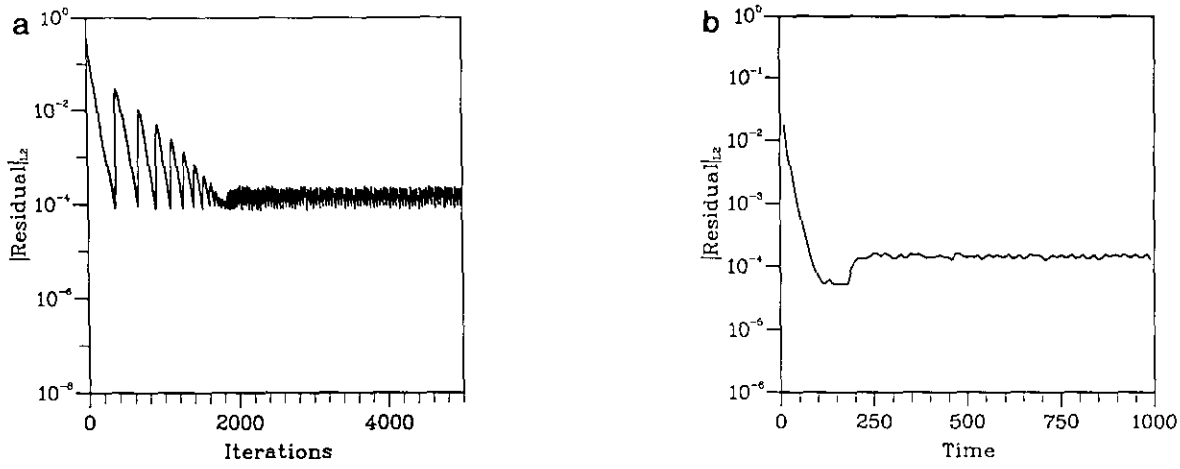

FIG. 4. Re = 1000.

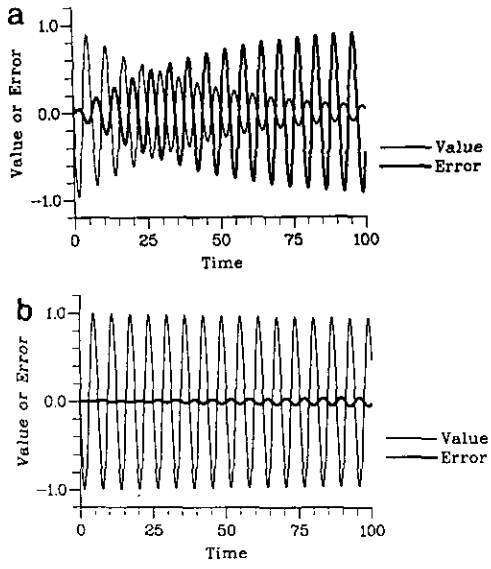

FIGURE 5

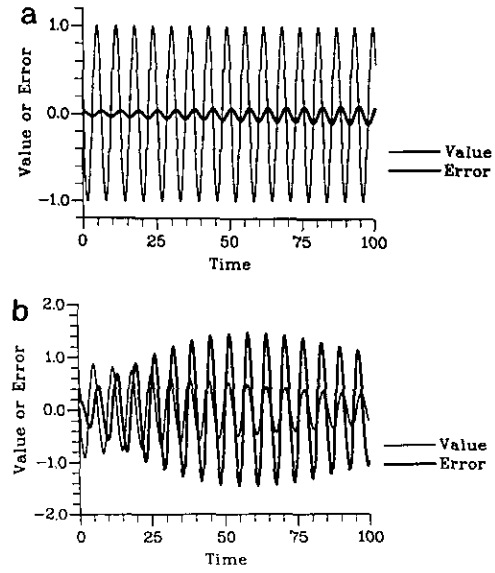

FIGURE 6

— when $\alpha\,\Delta\tau \geqslant 5$ CFL, the simplified form $\mathscr{L}_4$ can be used in place of the non-diagonal form $\mathscr{L}_2$ with a reduction of CPU time of roughly 40%.

— when $\mathrm{CFL}/10 < \alpha\,\Delta\tau < 5$ CFL, the simplified operators $\mathscr{L}_3$ and $\mathscr{L}_4$ either converge slowly or diverge. For this range of time steps, the non-diagonal form must be used.

These observations made in the driven cavity problem were confirmed in the free vortex shedding problem. Unfortunately, it was also observed that all the time steps of interest fell in the [CFL/10, 5 CFL] range. Therefore, all the computations described below were performed with the non-diagonal implicit operator $\mathscr{L}_2$.

Numerical tests done on the driven cavity problem also show that, when the unsteady procedure is used to compute steady-state solutions, the pseudo-steady problems have to be well converged if an accurate approximation of the steady solution is required. Precisely, if an acceptable approximation of the steady solution is characterized by

$$\|\mathscr{N}_h(U)\| < \varepsilon_1,$$

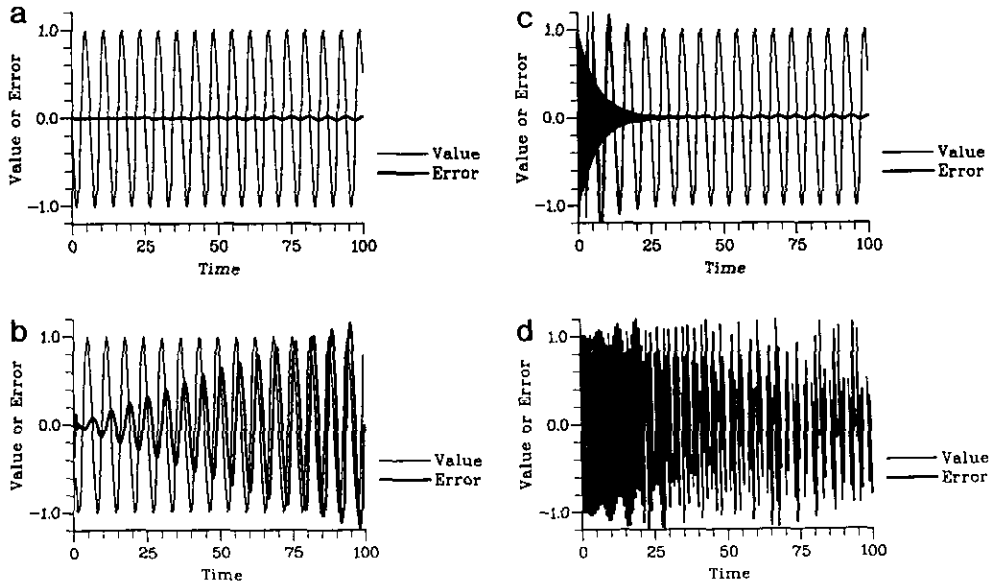

FIG. 7. Crank–Nicholson: (a) $DT = 0.05$, $k = 1$; (b) $DT = 0.4$, $k = 1$; (c) $DT = 0.04$, $k = 4$; (d) $DT = 0.4$, $k = 4$.

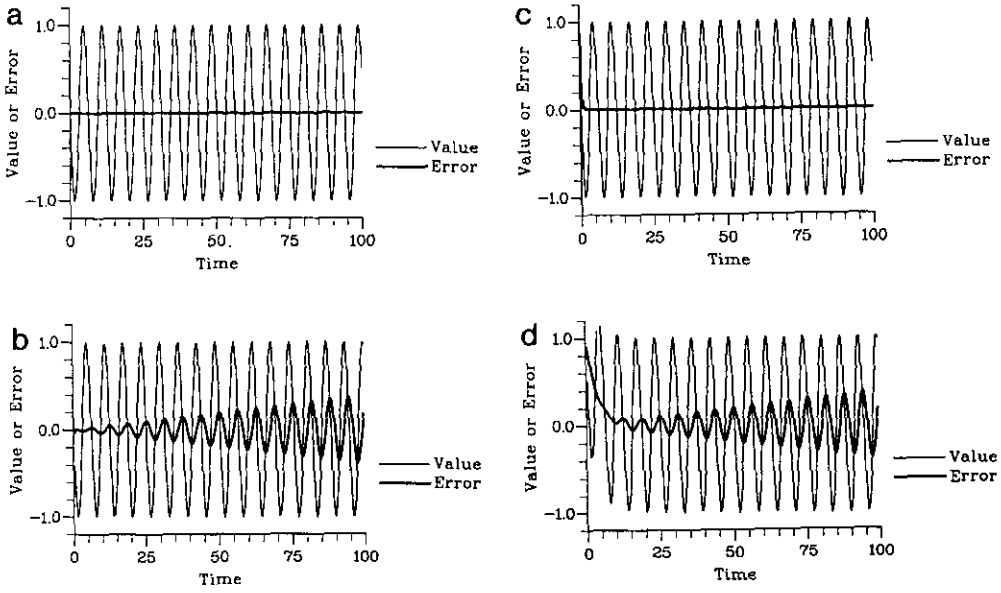

FIG. 8. Runge–Kutta: (a) $DT = 0.05$, $k = 1$; (b) $DT = 0.4$, $k = 1$; (c) $DT = 0.05$, $k = 4$; (d) $DT = 0.4$, $k = 4$.

then, it is imperative to solve the pseudo-steady problems accurately,

$$\|\alpha \tilde{I} U + \mathcal{N}_h(U) - \Phi^n\| < \varepsilon_2,$$

with

$$\varepsilon_2 \leqslant \varepsilon_1.$$

This property is illustrated in Fig. 4. The simulation was done with $\varepsilon_2 = 10^{-4}$. From Fig. 4b it is clear that the norm of the residual $\|\mathcal{N}_h(U)\|$ cannot be driven to machine zero; after the initial reduction, $\|\mathcal{N}_h(U)\|$ stabilizes around $\varepsilon_1 = O(\varepsilon_2)$.

4.3. Model Problem

The numerical properties of the various schemes described in Section 2 were first evaluated on a model problem proposed by Rannacher [16]. The problem to be solved is

$$\frac{du}{dt} + Au(t) = 0, \qquad \text{with} \quad u(0) = 0.$$

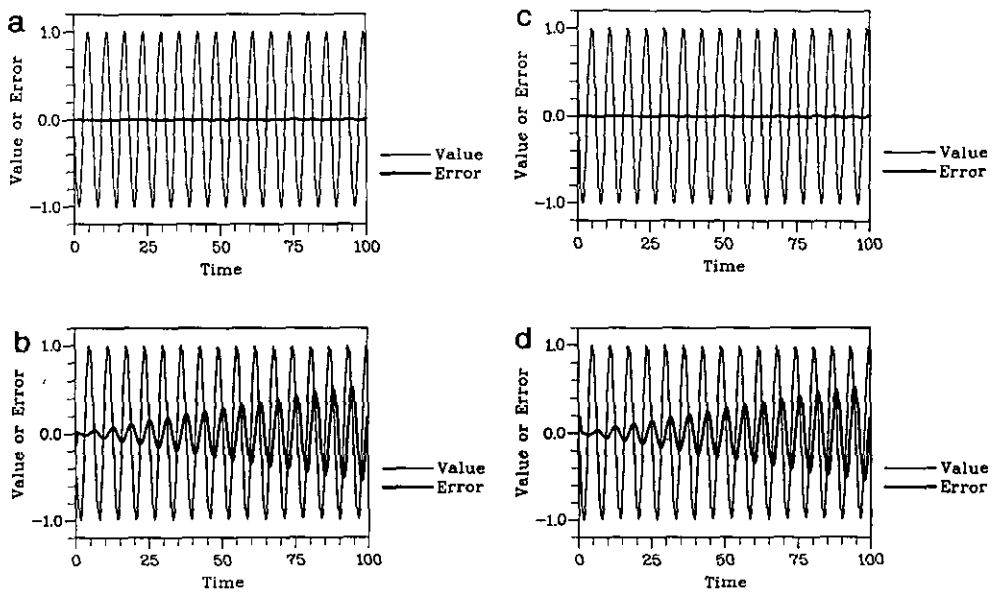

FIG. 9. Runge–Kutta: (a) $DT = 0.05$, $k = 1$; (b) $DT = 0.4$, $k = 1$; (c) $DT = 0.05$, $k = 4$; (d) $DT = 0.4$, $k = 4$.

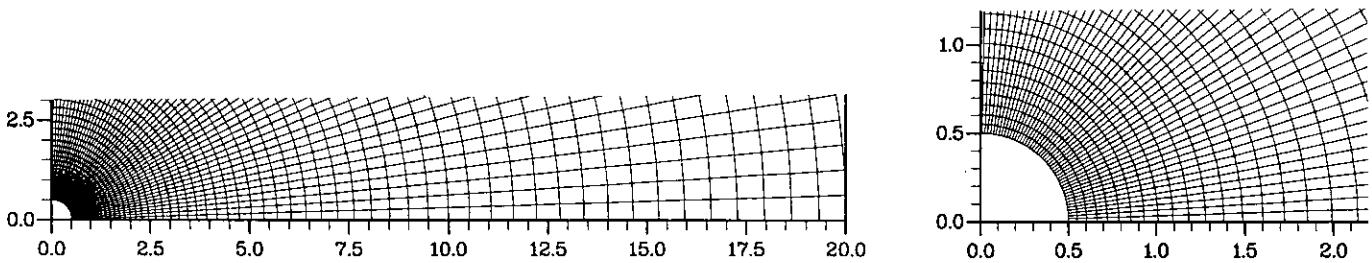

FIGURE 10

A is a 4×4 matrix having $\lambda_1 = 10^k$, $\lambda_2 = i$, $\lambda_3 = -i$, $\lambda_4 = 1$ as eigenvalues. Hence, the solution contains two periodic components and two decaying components. Also, the stiffness of the system can be changed by a modification of the value of k. The major properties of the different schemes can be seen from Figs. 5–9. In these figures, the displayed solution corresponds to the sum of the eigenvectors associated to λ_1 and λ_2. A comparison of the solutions shows that, as expected, the implicit Euler scheme solves the problem with acceptable accuracy only for very small time steps. Since the period of the signal is $T = 2\pi$, $\Delta t = 0.001$ corresponds to using ~ 6283 iterations to integrate the signal over one period. In comparison, the respective numbers for $\Delta t = 0.05$ and $\Delta t = 0.4$ are ~ 126 and ~ 15.7 time steps per period. The stiffness of the system has, on the other hand, no effect on the accuracy of the solution. This insensitivity is also observed with the backward-differencing scheme and results from the strong A-stability property. The solutions computed with the backward-differencing scheme are much more accurate, even if large errors can be observed for relatively large time steps. When comparing the backward-differencing scheme with the other second-order schemes, it is noted that the backward-differencing scheme is the most dissipative and least accurate scheme. The Crank–Nicolson scheme is non-dissipative; only dispersion errors are created by the scheme. These errors may be quite large when the time step is too large. But more noticeably, the Crank–Nicolson scheme is very sensitive to the k parameter, and clearly, it has difficulties in resolving stiff problems. The DIRK 2.1 scheme is the most accurate scheme considered in this study. The errors obtained with this scheme are even acceptable for $\Delta t = 0.4$. From Fig. 8d, it can be observed that, with this large time step, the stiff problem ($k = 4$) is not well resolved, since the scheme produces significant errors—on the solution associated with the eigenvalue λ_1—initially. These errors, however, do not destabilize the simulation as with the Crank–Nicolson scheme. With the DIRK 2.2 scheme, some errors are generated near $t = 0$, but, they are rapidly damped, Fig. 9d. On the other hand, for the periodic signal, the DIRK 2.2 scheme appears to be less accurate than DIRK 2.1.

In the remaining tests, it will be shown that the characteristics of the different integration schemes observed on this simple model problem have some repercussions on the simulation of unsteady flows. For instance, the ability of the scheme to solve correctly stiff problems is directly linked to the number of iterations needed to solve the pseudo-steady problems. This simple model problem is therefore a good test for evaluating integration schemes for the solution of the unsteady Navier–Stokes equations.

In the following series of calculations, the integration schemes will be extensively tested on unsteady flow problems. Three aspects have been specifically investigated: (i) the accuracy of the different integration schemes; (ii) the

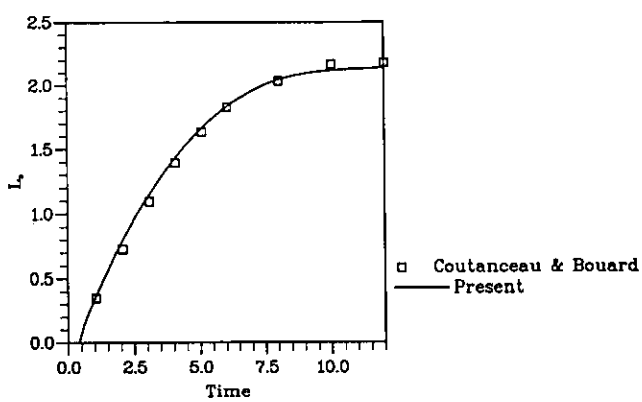

FIGURE 11

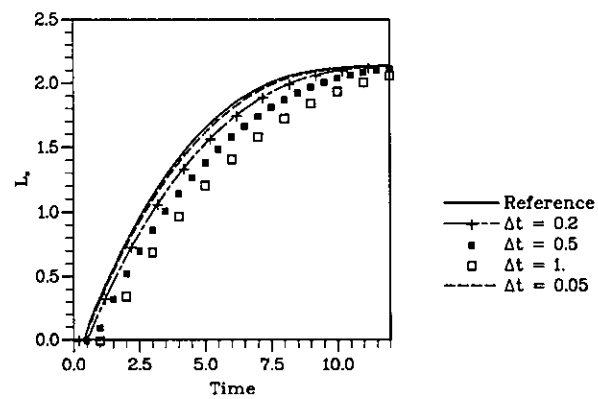

FIGURE 12

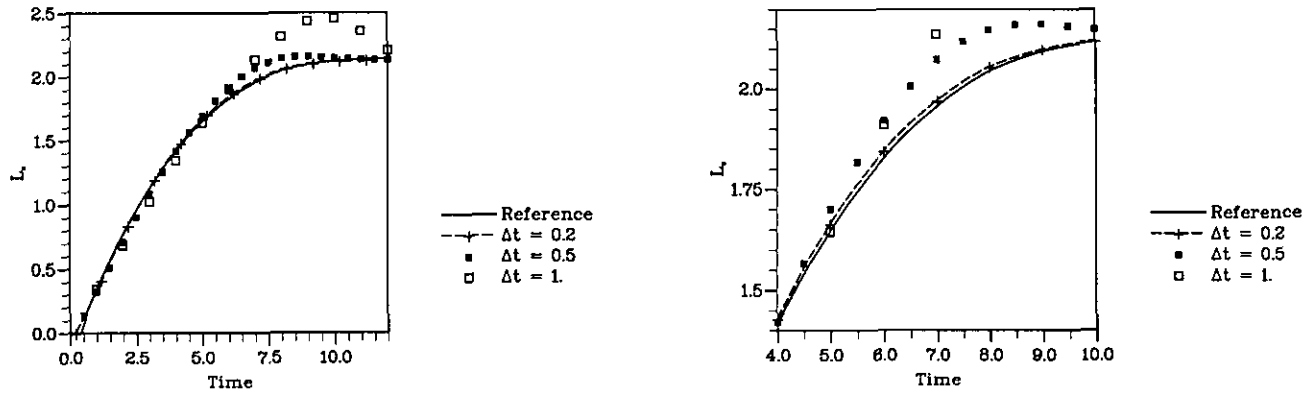

FIGURE 13

number of iterations required for each method to solve the pseudo-steady problems; (iii) the influence of the convergence level imposed on the solution of the pseudo-steady problems. These effects were studied with reference to two flow problems. The flow-field behind an impulsively-started cylinder at $\mathrm{Re} = 40$ was used to investigate the properties of the schemes for the simulation of transient flow-fields. The ability of the integration schemes to model properly periodic motions have been evaluated on the fully developed wake emanating from a circular cylinder at $\mathrm{Re} = 200$.

4.4. Impulsively Started Cylinder, Re = 40

For this Reynolds number, the wake created by a moving cylinder remains symmetric and attains a steady state. The present simulations are focused on the unsteady motion generated just after the cylinder has been impulsively started. The unsteady computations were performed on a 100×60 cylindrical mesh, Fig. 10, generated around the half-cylinder. The outer boundary was located 20 diameters from the center of the cylinder. Characteristic boundary conditions were imposed on the open boundary; i.e., the Roe–Riemann solver (7) was employed at the boundary using the free-stream conditions as an outer flow-field. Such approximation is reasonable outside of the wake. In the wake region this condition is inappropriate and, if used, will affect the solution. However, if the outer boundary is located sufficiently far from the cylinder, its effects on the converged solution will be negligible.

4.4.1. *Validation*

To validate the code, the evolution of the separation length of the recirculation has been compared against experimental results. The agreement is satisfactory, Fig. 11. The separation length is slightly under-estimated for $t \geqslant 10$. This may be attributed to effects of the boundary condition imposed in the computations. It is also noted that boundary effects may also be present in the measurements. Another possible source of error is the mesh size. With the clustering of the grid points used, the size of the cell grows away from the cylinder. Therefore, as the separation length increases, its determination cannot be better than the "local" grid size. The discrepancies observed in Fig. 11 are, however, small and comparable to the ones obtained from other numerical simulations [3, 27]. The numerical results shown in Fig. 11 were computed with the DIRK 2.1 scheme using a non-dimensional time step of $\Delta t = 0.2$. This solution will be used as reference solution in the remaining computations.

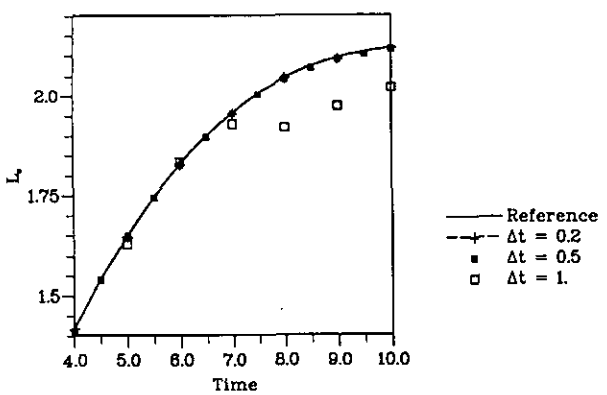

FIGURE 14

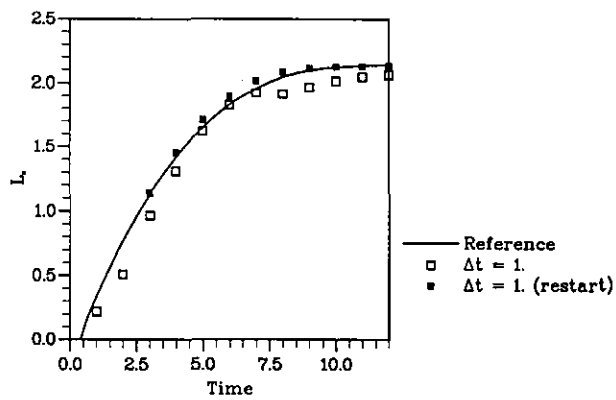

FIGURE 15

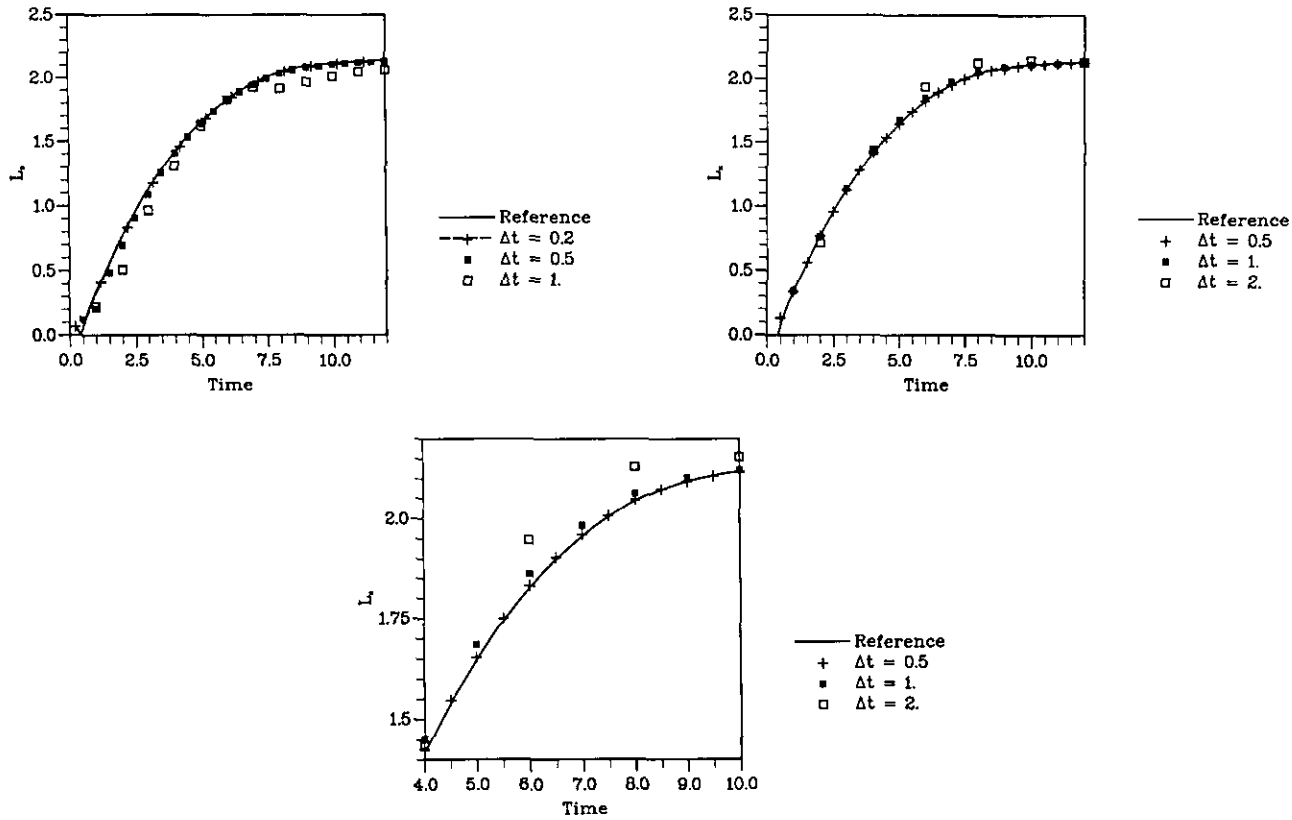

FIGURE 16

4.4.2. *Influence of the Integration Schemes on Accuracy*

The initial fields obtained using the procedure described in Section 4.1 are employed to start several computations using the various integration schemes and different time steps. Because two time levels are needed for updating the flow-fields with the backward-differencing scheme, a special initialization has to be employed in this case. All the computations presented in this paper with the backward-differencing scheme use the DIRK2.1 scheme at the first iteration. The results obtained with the different integration schemes are presented in Figs. 12–17. For this flow problem, the integration schemes behave exactly as for the model problem. The Euler scheme, even with a very small time step, $\Delta t = 0.05$, does not predict the correct evolution. On the other hand, the evolution of the separation length computed with the DIRK2.1 scheme using $\Delta t = 0.2$, reference solution, and $\Delta t = 0.5$ are identicals. (This justifies the use of these results as "the reference solution.") Comparisons between the different solutions shows that the most accurate

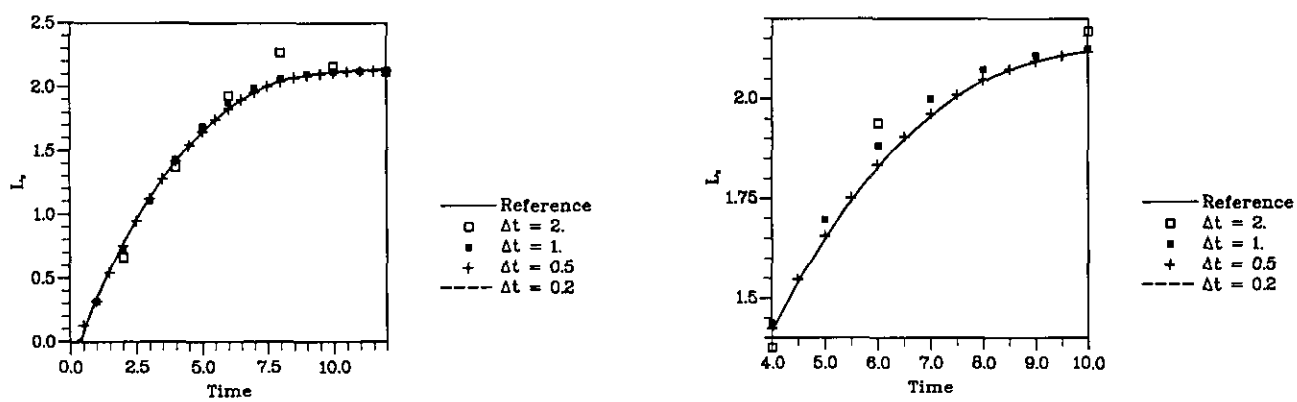

FIGURE 17

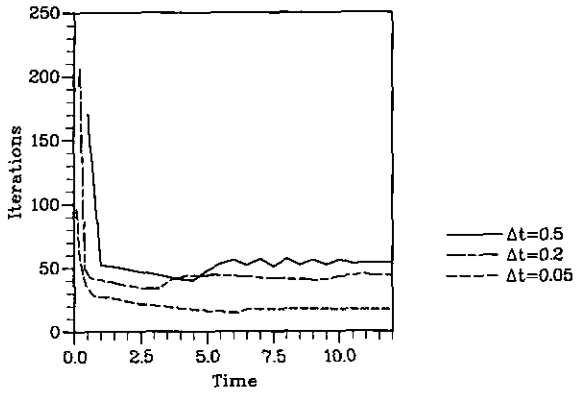

FIG. 18. Implicit Euler.

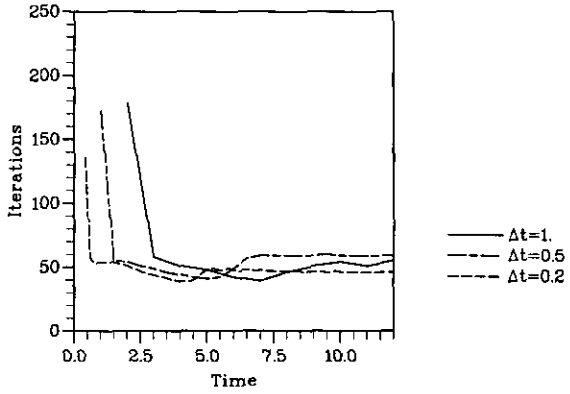

FIG. 19. Backward-differencing.

scheme is the DIRK 2.1 scheme followed by the DIRK 2.2, the Crank–Nicolson, the backward-differencing, and the Euler scheme. It may be noted that with each second-order scheme the "correct" solution is obtained with $\Delta t = 0.2$. The Crank–Nicolson scheme has a peculiar behavior for $\Delta t = 1$. As shown by the computations on the model problem, the Crank–Nicolson scheme has difficulties to capturing rapid transient motions when the time step is too large. Therefore, it was suspected that the strange behavior results from the singularity at $t = 0$. This conjecture is supported by the unexpected behavior of the Crank–Nicolson scheme near $t = 0$ with $\Delta t = 0.2$. To confirm this suspicion, the following test was conducted. The flow-field was integrated from $t = 0$ to $t = 3$ using the DIRK 2.1 scheme before starting the computations with the Crank–Nicolson scheme with $\Delta t = 1$. The influence of the starting condition on the Crank–Nicolson scheme with $\Delta t = 1$ is illustrated in Fig. 15. Clearly, when the influence of the initial transients ($t = 0$ to $t = 3$) is

4.4.3. *Investigation on the Efficiency of Integration Schemes*

For the different integration schemes, the evolution of the number of iterations needed to converge the residual of the pseudo-steady system below $\varepsilon = 10^{-4}$ is represented as a function of flow time in Figs. 18–22. The corresponding total number of iterations and the CPU time (measured on a Cray YMP) required for the integration of the incompressible Navier–Stokes equations from $t = 0$ to $t = 12$, is given in Table I. To assess the efficiency of the upwind approximate factorization scheme, the CPU time needed for solving the *steady* equations has also been indicated in Table I. Comparing this CPU time with the averaged time necessary for solving the pseudo-steady problems when the unsteady equations are integrated with a large time step ($\Delta t = 1$), it is noted that only half of the CPU time required for solving the steady equations is needed for solving the pseudo-steady equations with a strongly A-stable scheme

TABLE I

Execution Characteristics on the Impulsively Started Cylinder Problem, Re = 40

Scheme	Δt	Pseudo CFL number	Total number of iterations	Run CPU time (s)
Implicit Euler	1	20	684	35
	0.5	30	1343	69
	0.2	40	2652	136
	0.05	60	5300	274
Backward-differencing	1	25	1145	58
	0.5	40	1790	92
	0.2	70	3437	177
Crank–Nicolson	1	40	2074	106
	0.5	50	6420	324
	0.2	60	22156	1130
DIRK 2.1	2	30	1950	99
	1	40	2375	121
	0.5	80	5803	296
	0.2	100	13929	710
DIRK 2.2	2	30	976	50
	1	40	1368	70
	0.5	80	2231	115
	0.2	90	4072	209
Steady calculations		25	124	6.4

time for each solution of the pseudo-steady problems. An examination of Figs. 18–22 shows that the strongly A-stable schemes can easily be recognized. For instance, the behavior of the implicit Euler and of the backward-differencing schemes are almost identical. Apart from the first time step that necessitates more iterations, the number of iterations needed for converging the pseudo-steady system settle down to around 50. This value seems to be insensitive to the time step Δt. The DIRK 2.2 scheme differs slightly from the other two strongly A-stable schemes by the larger influence of the time singularity at $t = 0$. For $\Delta t = 2$, the singularity affects the convergence of the pseudo-steady system up to

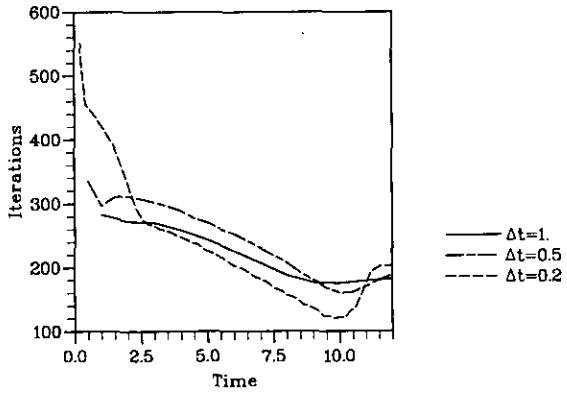

FIG. 21. DIRK 2.1.

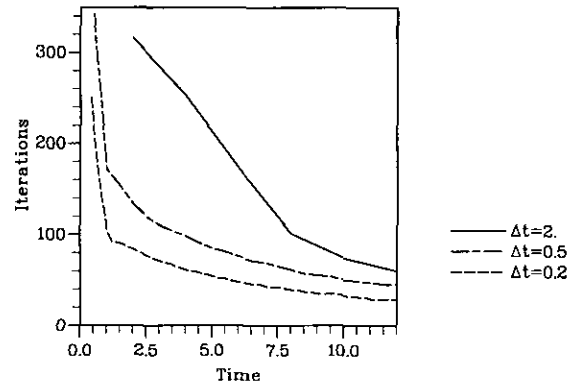

FIG. 22. DIRK 2.2.

approximately $t = 8$. From Fig. 22 it can be noted that this influence, however, diminishes with a reduction of the time step. Since the convergence histories observed with the implicit Euler and backward-differencing schemes are almost identical, the CPU time necessary for the whole simulation should be also nearly the same. In Table I this CPU time is, however, different. The difference is caused primarily by the DIRK 2.1 scheme used at the first iteration when the simulation is conducted with the backward-differencing scheme. From Table I it can be seen that the DIRK 2.2 scheme needs also a few more iterations for the whole integration. The increase in CPU time is nevertheless small, especially when considering that with the DIRK 2.2 scheme two stages must be computed per time step and is largely compensated by the improvement obtained in the accuracy. The DIRK 2.1 scheme requires more than twice as many iterations than the DIRK 2.2 scheme for a slight

TABLE II

Execution Characteristics on the Impulsively Started Cylinder Problem

Scheme	Δt	Pseudo CFL number	Total number of iterations	Run CPU time (s)
Backward-differencing	0.5	40	1636	84
	0.2	70	3450	177
Crank–Nicolson	1	40	892	46
	0.5	50	2074	107
	0.2	60	7084	363
DIRK 2.1	2	30	846	43
	1	40	1809	93
	0.5	80	3812	195
	0.2	100	11555	588
DIRK 2.2	2	30	599	30
	1	40	1006	51
	0.5	80	2140	108
	0.2	90	4019	204

Note. Re = 40; uniform initial flow-field.

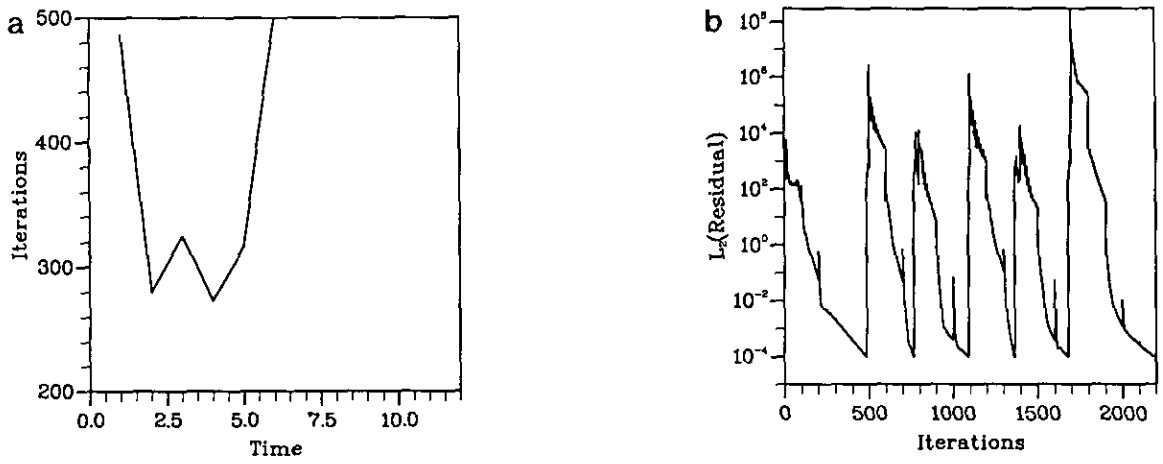

FIG. 23. Crank–Nicholson, 100×60 mesh, $\Delta t = 1$.

improvement in accuracy. Therefore, the use of a scheme strongly dissipative for $\Re(\lambda) \to +\infty$ appears to have a significant influence on the convergence properties of the method. This finding is further supported by the behavior of the Crank–Nicolson scheme. This scheme is not dissipative for $\Re(\lambda) \to +\infty$. As it can be observed in Fig. 20, the convergence history obtained with the Crank–Nicolson scheme is completely different from the other schemes. It is surprising that the number of iterations needed to solve the pseudo-steady system increases with time and increases also when the time step is reduced. In consequence, the CPU time required for the simulation is much larger than for the other schemes, making the procedure quite inefficient. Furthermore, it has been found that with the Crank–Nicolson scheme, it was necessary to update the artificial compressibility parameter c^2—according to Eq. (6)—at each iteration. If the update is performed only every 100 iterations, Fig. 23, the integration of the Navier–Stokes equations diverges at $t = 6$. With the other integration schemes, the frequency used for the update of c^2 was found to have only a minor influence on the convergence history of the procedure.

The great sensitivity of the integration schemes that are not able to damp well the fast decaying motions, is shown in Table II. In this table, calculations similar to the previous calculations are reported. The only difference is the initial solution. Instead of integrating the incompressible Navier–Stokes equations with a divergence-free initial field, the equations were integrated from a uniform flow-field. When comparing the results given in Tables I and II, it is apparent that the schemes which damp least, the fast decaying modes are the most susceptible to the influence of the initial conditions. For the strongly A-stable schemes, i.e., for the backward-differencing and the DIRK 2.2 schemes, no difference is found for small time steps. For larger time steps some influence is observed. Since the DIRK 2.1 scheme is less strongly A-stable than the previous schemes, the simulation is also more affected by the initial conditions. For instance, even with $\Delta t = 0.2$, the CPU time necessary for the simulation performed with the "unsuitable" initial condition is approximately 20% less than with the proper initial condition. For the Crank–Nicolson scheme, the reduction is considerable. With the "unsuitable" initial condition, the required CPU time is three times less than the time needed with the realistic-flow initial condition. Again, it is found that the time step dependence of the convergence properties of the Crank–Nicolson scheme, are the reverse of the dependence observed with the other integration schemes; the CPU reduction obtained with the "unsuitable" initial condition diminishes with larger time steps.

Finally, the computing effort necessary for obtaining an accurate solution are compared in Table III. The relative efficiency of the different schemes may thus be assessed. It should be noted out that the figures listed in Table III should be viewed as *indicative* of relative merits. There has been some attempt to optimize the pseudo-CFL number of the computations, but a slight variation in the CFL number may still influence the computational time. Second, the

TABLE III

Quality Characteristics of the Integration Schemes

Scheme	Δt	CPU time	Error
Implicit Euler	0.05	274	2.2×10^{-2}
Backward-differencing	0.2	177	1.1×10^{-2}
Crank–Nicolson	0.5	324	9.6×10^{-3}
DIRK 2.1	0.5	296	2.9×10^{-3}
	1	121	1.8×10^{-2}
DIRK 2.2	0.5	115	3.0×10^{-3}
	1	70	2.5×10^{-2}

Note. Impulsively started cylinder problem, $\mathrm{Re} = 40$.

error reported in Table III is a rough estimate. It was computed according to

$$\text{error} = \frac{1}{N} \sum_{\mathscr{S}} |l_{ref} - l_{sch}|,$$

where

— l_{ref} is the recirculation length computed with the DIRK 2.1 scheme with $\Delta t = 0.2$,

— l_{sch} is the recirculation length computed with the different schemes reported in Table III,

— $\mathscr{S}$ is the set of points for which l_{ref} and l_{sch} are both known at the same time,

— N is the number of points of the set $\mathscr{S}$.

Keeping in mind the above remarks, it may be concluded from Table III that DIRK 2.2 is the best scheme considered in the present study. It requires approximately half the CPU time for a given accuracy than the second best schemes, i.e., the DIRK 2.1 and backward-differencing schemes. The implicit Euler scheme that was judged very poorly by considering only the accuracy of the scheme is now found to be better than the Crank–Nicolson scheme. However, it may be remarked, that with exactly the same numerical procedure, an integration with the implicit Euler scheme necessitates roughly four times as much CPU time as the optimized DIRK 2.2 scheme.

4.4.4. *Influence of the Convergence Criterion*

A series of calculations on the flow behind an impulsively started cylinder were used to study the influence of the tolerance level on the solution efficiency of the pseudo-steady system (4). Two tolerance levels were employed: $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-1}$. As described in Section 2.2, with a given integration scheme several time corrections can be employed to determine the new fields U^{n+1}. For multi-stage schemes, other corrections can be devised. For instance, the correction procedure can be applied at the end of the last stage and not after each intermediate stage. When the "correction (O)" procedure is employed, the procedure will be referred to as the "correction (HC)" procedure. It may be noted that with the "correction (N)" procedure, the correction is done only after the last stage. The "correction (NN)" procedure differs from the "correction (N)" by the initial field employed at the beginning of the second stage. In the "correction (N)" procedure, this field is $U^{(2),0} = \bar{U}^{(1)}$, whereas in the "correction (NN)" procedure the initial field is taken as $U^{(2),0} = \bar{U}^{n}$. As indicated in Table IV this variation may change the number of iterations needed to solve

TABLE IV

Influence of the Precision Used to Solve the Pseudo-steady System

Scheme	ε	Total number of iterations	Run CPU time (s)	Correction
Backward-differencing	10^{-4}	3437	177	—
	10^{-1}	2373	121	corr (O)
	10^{-1}	364	20	corr (N)
	10^{-1}	195	10	no corr
Crank–Nicolson	10^{-4}	22156	1130	—
	10^{-1}	5296	270	corr (O)
	10^{-1}	1994	103	corr (N)
	10^{-1}	1979	102	no corr
DIRK 2.1	10^{-4}	13929	710	—
	10^{-1}	10768	557	corr (O)
	10^{-1}	5085	261	corr (HC)
	10^{-1}	2344	123	corr (N)
	10^{-1}	2802	147	corr (NN)
	10^{-1}	1718	87	no corr
DIRK 2.2	10^{-4}	4072	209	—
	10^{-1}	6607	339	corr (O)
	10^{-1}	3429	176	corr (HC)
	10^{-1}	1042	55	corr (N)
	10^{-1}	1041	55	corr (NN)
	10^{-1}	762	39	no corr

Note. Impulsively started cylinder problem, Re = 40.

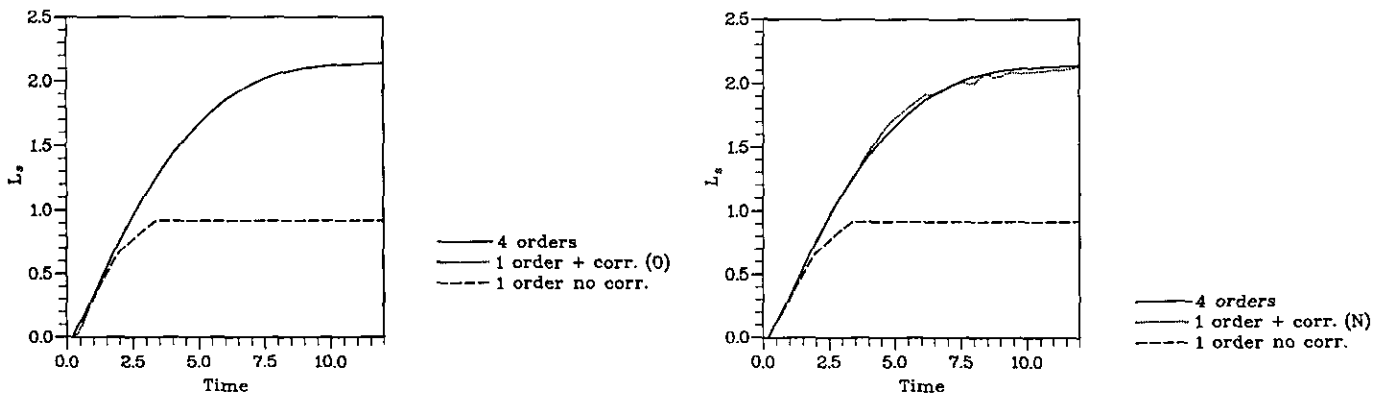

FIGURE 24

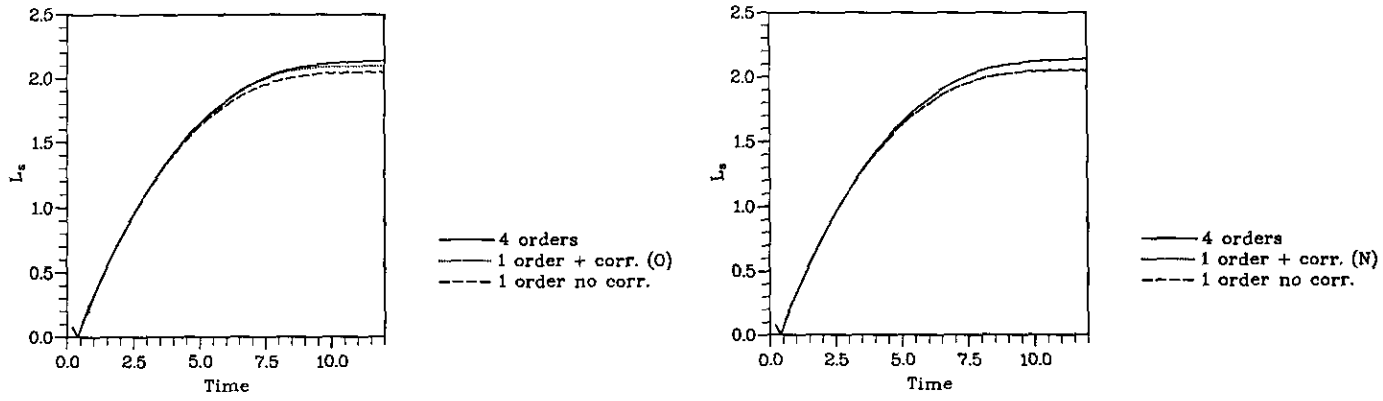

FIGURE 25

the system (4) and in consequence, it may modify the solution.

From the time evolutions of the recirculation length presented in Fig. 24, the suitability of the "correction (O)" procedure seems confirmed. A more representative measure of efficiency should account for the amount of computational work. By examining the accuracy obtained with the different methods, Figs. 24–27 and the work necessary for each procedure, Table IV it can be noted that the recirculation length is correctly determined when a sufficient number of iterations are performed, e.g., a backward-differencing scheme with $\varepsilon = 10^{-1}$ and "correction (O)," Fig. 24. When a similar number of iterations are used, almost identical solutions are obtained: Crank–Nicolson scheme with $\varepsilon = 10^{-1}$ without correction and $\varepsilon = 10^{-1}$ plus "correction (N)", Fig. 25. It is therefore important to consider overall efficiency rather than formal accuracy of correction procedures. It might be argued that their efficiency is due to the

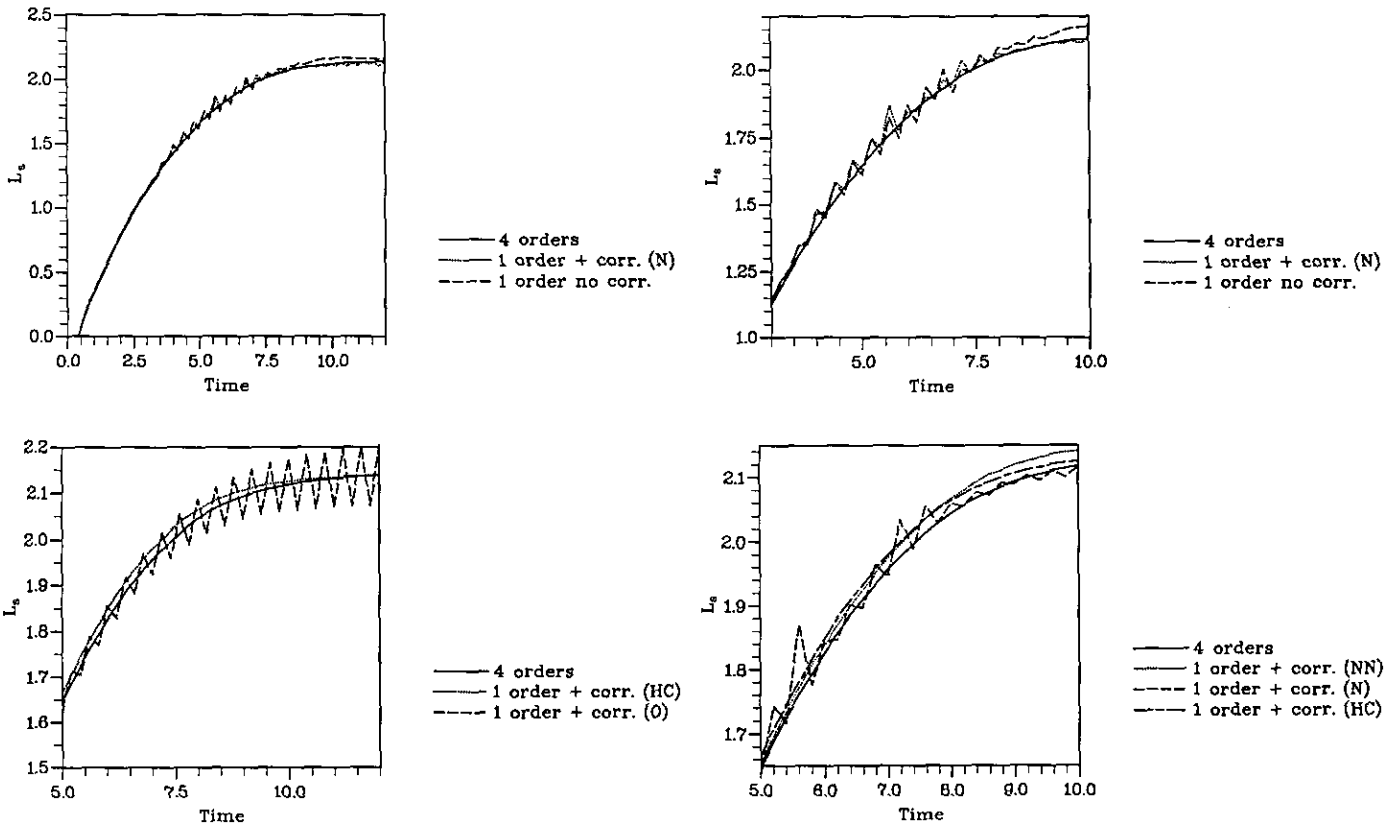

FIGURE 26

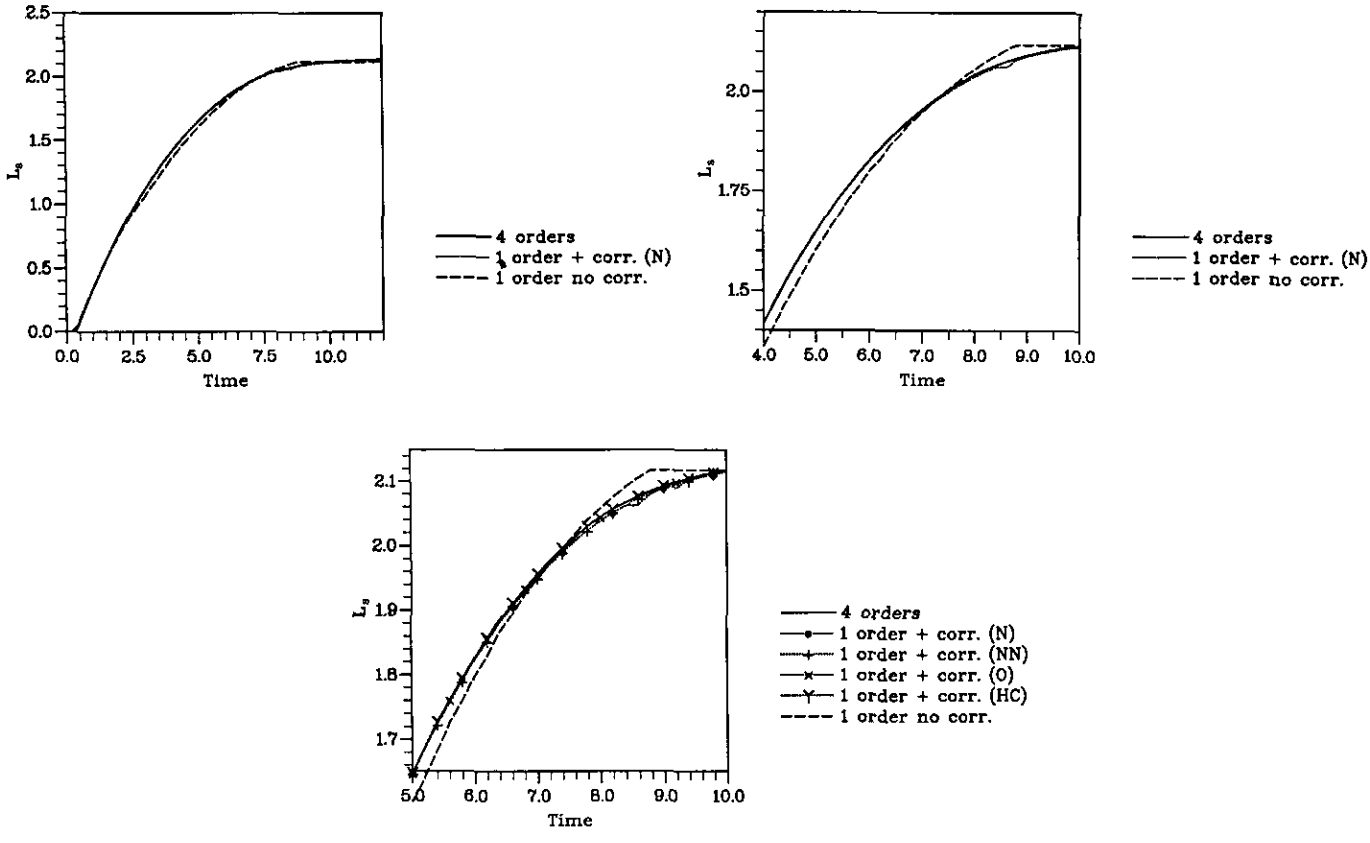

FIGURE 27

greater number of iterations necessary to solve the system (4). Whereas 195 iterations are only necessary to converge the pseudo-steady system with the backward-differencing scheme, $\varepsilon = 10^{-1}$, and without correction, this number jumps to 2373 iterations with the "correction (O)" procedure. Thus only 30% fewer iterations are necessary to solve the system to $\varepsilon = 10^{-1}$ with "correction (O)" than to solve accurately the system (4) with $\varepsilon = 10^{-4}$. With the "correction (O)" procedure, it can be observed from Table IV that, except for the Crank–Nicolson scheme, the reduction in CPU time is always moderate; for the DIRK2.2 scheme, even more CPU time is required. With the "correction (N) or (NN)" procedures, the CPU-time reduction is more pronounced. The difference in the behavior observed with "correction (O)" and "correction (N)" procedures is probably caused by the instability created by the *explicit* fixed point integration. With the "correction (O)" procedure $\tilde{U}^{n+1}$ serves both as the initial solution for U^{n+2} and to estimate $\mathcal{N}_h(U^{n+1})$ in the computation of the source term Φ^{n+1}, whereas after the replacement of $\tilde{U}^{n+1}$ by its expression, $\tilde{U}^{n+1}$ is not used at all in the "correction (N)" procedure. Therefore, the *explicit* fixed point integration cannot destabilize the computations performed using the "correction (N)" procedure.

From Figs. 26, it can be observed that the DIRK2.1 scheme is very sensitive to the correction procedure used. The "correction (O)" procedure appears to "over-correct" the results, leading to visible oscillations in the computed recirculation length. The "correction (N)," on the other hand, does not improve significantly the results obtained without correction. In agreement with the previous comments, the work needed with "correction (N)" is also similar

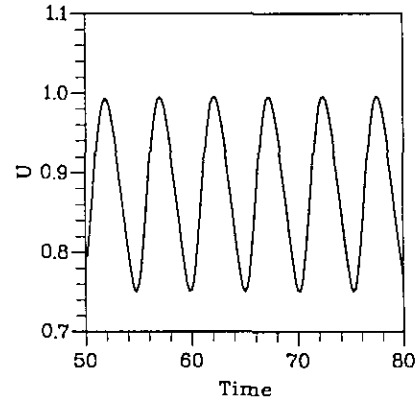

FIGURE 28

TABLE V

Computed and Measured Lift and Drag Coefficients and Strouhal Number

		C_D	C_L	St	Δt	R_∞	Mesh
Gerrard [23]	exp			0.18–0.20			
Wille [22]	exp	1.3					
Lecointe & Piquet [20]	comp	1.58 ± 0.0035	± 0.5	0.194	0.05	83	91×46
Rogers & Kwak [1]	comp	1.23 ± 0.05	± 0.65	0.185	0.025	10	100×60
Rosenfeld et al. [2]	comp	1.31 ± 0.04	± 0.65	0.20	?	50	85×81
Present	comp	1.39 ± 0.04	± 0.62	0.196	0.1	10	100×60
Present	comp	1.35 ± 0.04	± 0.62	0.195	0.2	62.8	200×120

Note. Vortex shedding problem.

to the work required without correction. The increased work necessary to solve the pseudo-steady system (4)—DIRK 2.1 scheme with $\varepsilon = 10^{-1}$ and "correction (NN) or (HC)" procedures—leads to better estimates for the recirculation length. With the DIRK 2.2 integration scheme, the recirculation length predicted with either correction procedure is accurate. Even with a large tolerance, $\varepsilon = 10^{-1}$, and without correction, an acceptable estimate of the recirculation length is found. Since the number of iterations required to converge the pseudo-steady system is significantly smaller with the "correction (N)" procedure than with the "correction (O)" procedure, the former method is to be preferred.

The many studies performed on the impulsively started cylinder at $\mathrm{Re} = 40$, show that the DIRK 2.2 scheme described in Section 2.1 offers the best compromis between accuracy, efficiency, and sensitivity (robustness).

4.5. Free Vortex Shedding, Circular Cylinder, $\mathrm{Re} = 200$

The different tests conducted on the impulsively started cylinder were reported on the periodic vortex shedding problem for which the importance of the fast decaying modes should be less pronounced. In the following, the vortex shedding was triggered to reduce the computational time since destabilizing by round-off errors would require long integration times. As in [3, 28], the von Kármán street is initiated by an asymmetric perturbation consisting of rotating the cylinder in clockwise and counterclockwise directions. Then, after the initial transient phase, a periodic motion is established.

4.5.1. *Validation*

All comparisons presented on the vortex shedding problem were done on a cylindrical 120×60 mesh with clustering of the points near the body. The other boundary was located 10 diameters away from the center of the cylinder. Free-stream conditions were applied at this boundary in characteristic form. As shown in [29], these conditions should be preferred to primitive-variable boundary conditions because they eliminate spurious secondary frequencies (not multiples of the Strouhal). Indeed, in all the computations performed on the circular cylinder at $\mathrm{Re} = 200$, no evidence of such secondary frequencies has been found. To check the results computed on the 120×60 mesh further,

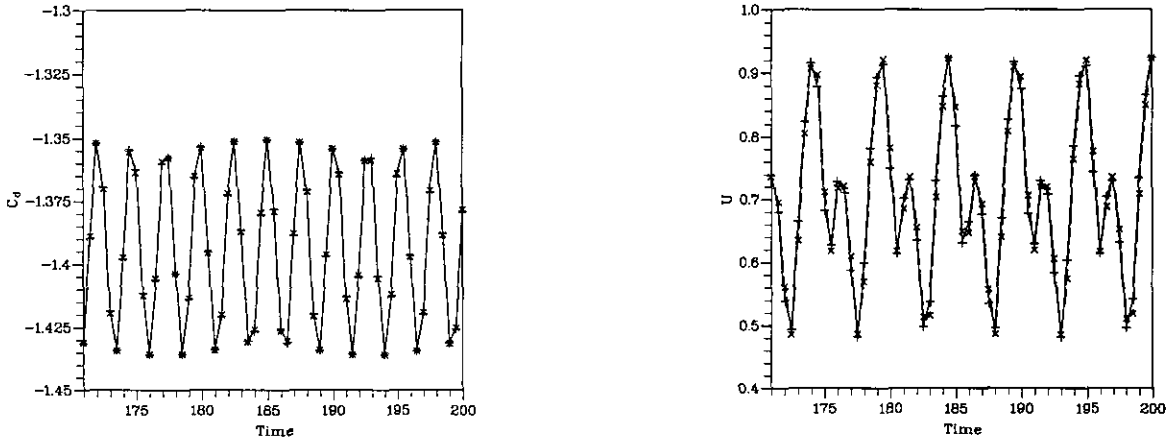

FIGURE 29

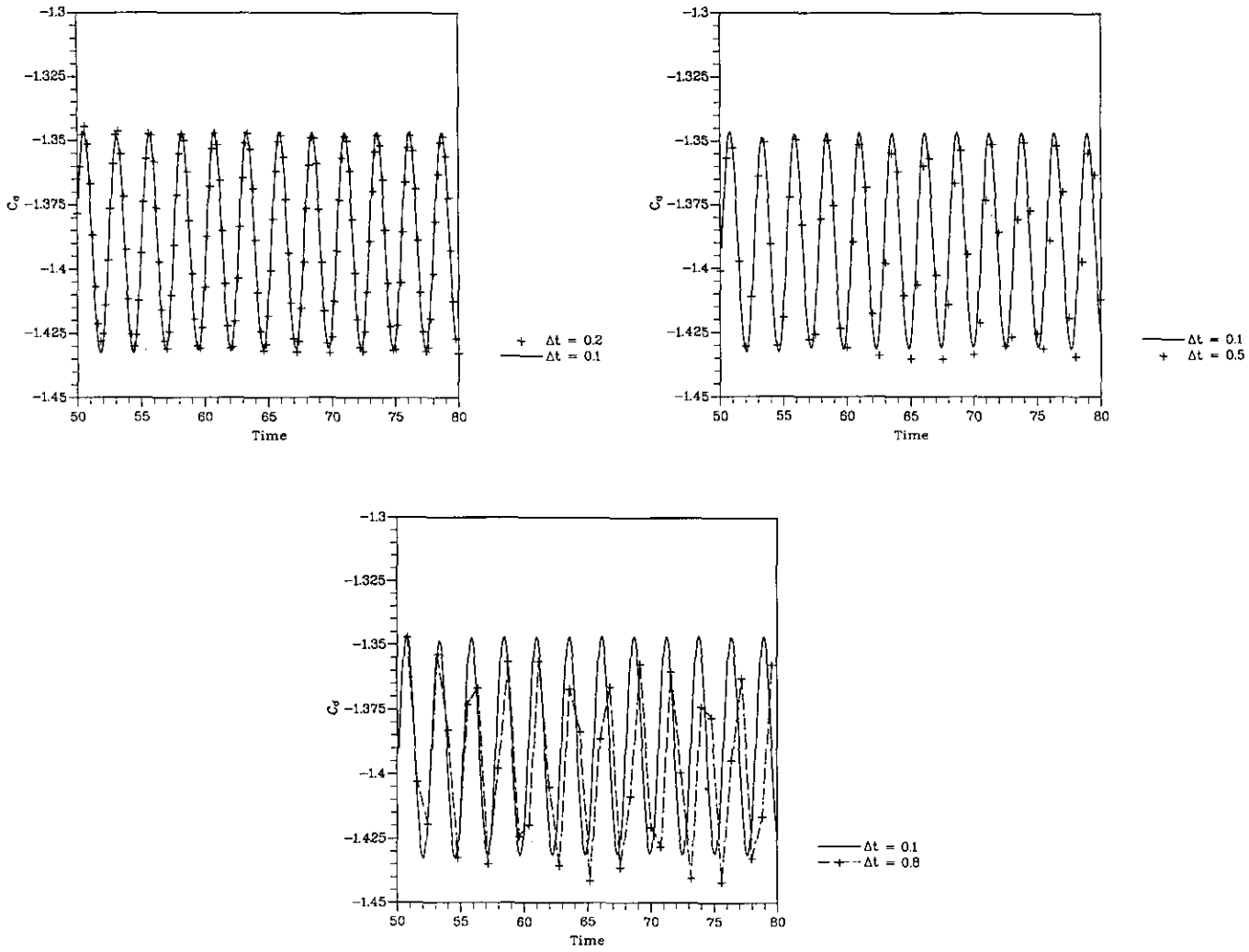

FIGURE 30

the vortex shedding problem was simulated on a finer mesh. This is composed of 200×120 grid points and the outer boundary was located at $R_\infty = 62.8$. The Strouhal number, the lift and drag coefficients calculated on the two meshes are given in Table V. In this table, other calculations [2, 3, 28] and measurements [30, 31] are also reported. The flow parameters obtained on both grids are almost identical, Table V. As observed also in [26], the mesh density is sufficient. Finally, with the DIRK 2.1 scheme using a time step of $\Delta t = 0.5$, the flow-field generated behind the cylinder was simulated for longer times to check the repeatability of the signal. From Fig. 29, the repeatability is clearly established; hence, it is possible to claim with some confidence that confinement effects are not obviously affecting the simulations. These numerical experiments show that the 100×60 grid can be used to test the various integration

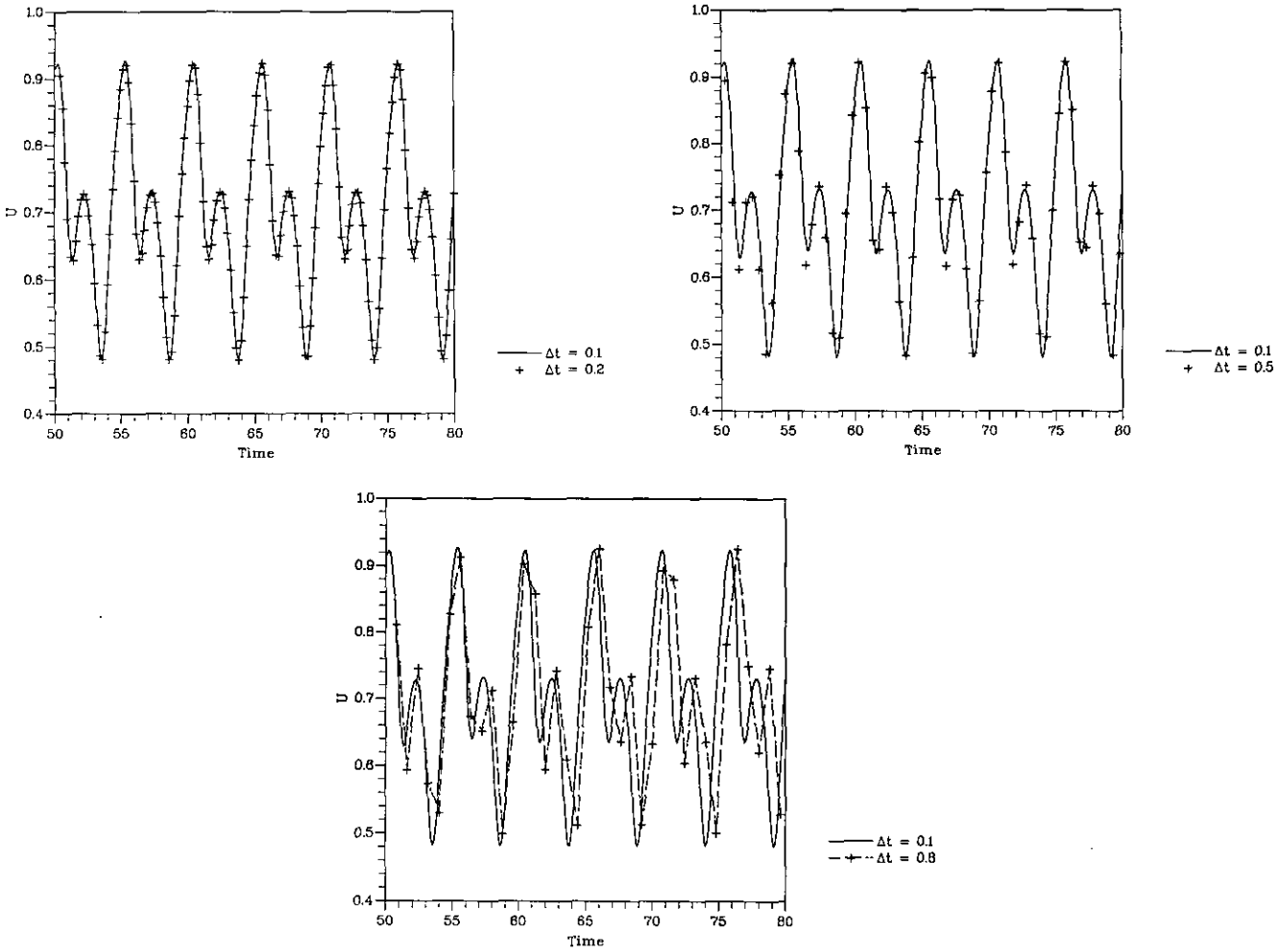

FIGURE 31

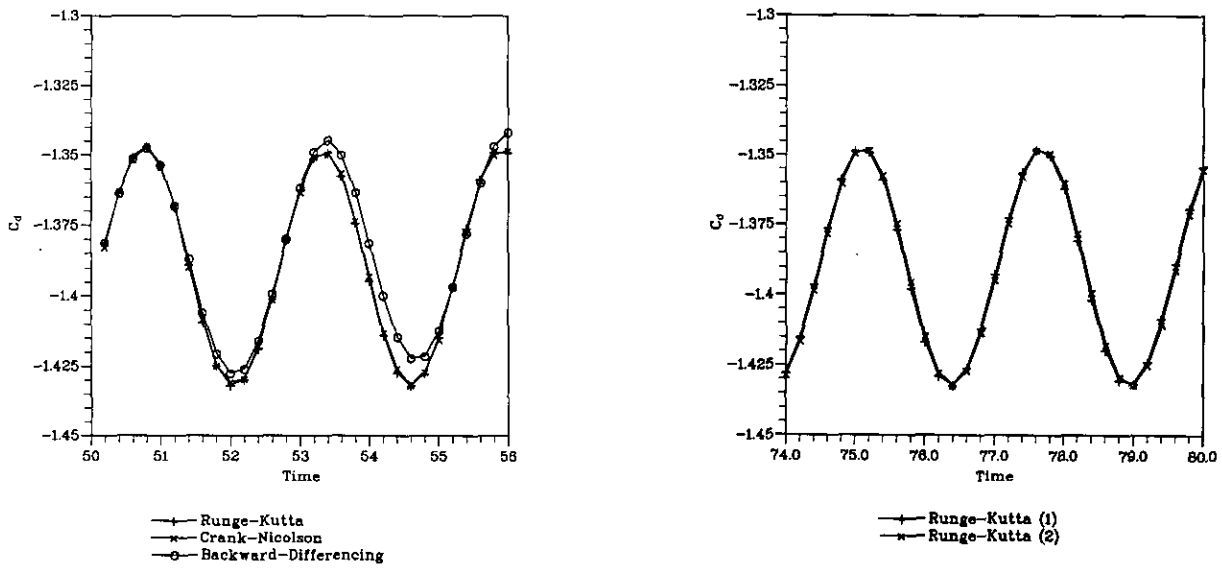

FIGURE 32

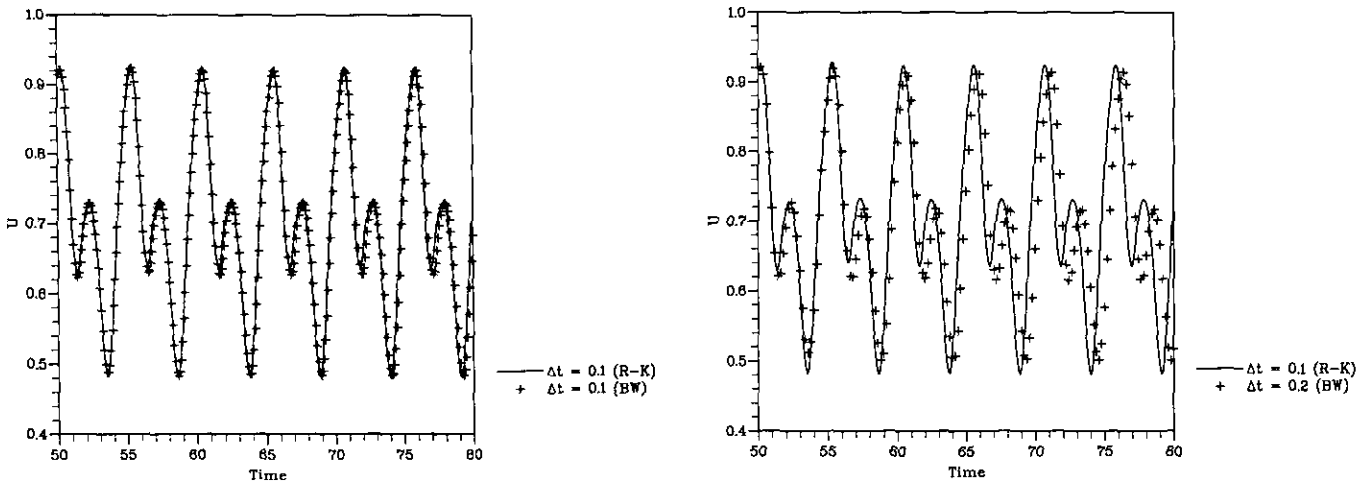

FIGURE 33

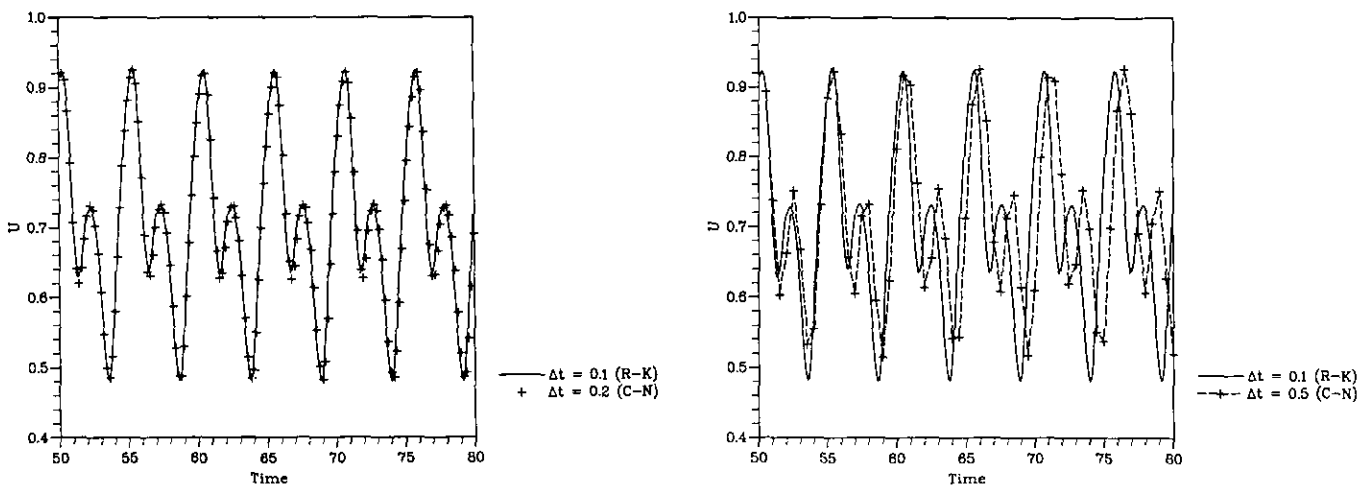

FIGURE 34

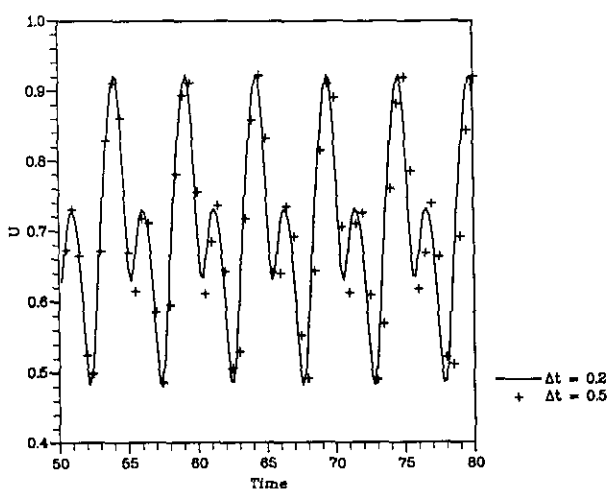

FIGURE 35

performed with time steps $\Delta t = 0.1$ and $\Delta t = 0.2$. The solution computed using the DIRK2.1 scheme will thus be designated as the reference solution. It can be observed that even with a very large time step, $\Delta t = 0.8$, the qualitative behavior of the solution is well reproduced, Figs. 30 and 31. It should be pointed out that with $\Delta t = 0.8$, signals with the lowest frequency are sampled only six times.

In Figs. 32–35, the solutions computed using the various integration schemes with different time steps are given. Two signals were analyzed: an integral quantity, the drag coefficient, and a pointwise quantity, the streamwise velocity at a point in the near wake, and 2.5 diameters behind the cylinder. From Fig. 32, it can be observed that only the backward-differencing scheme needs better time resolution to follow the drag coefficient signal. In this case, even with a time step of $\Delta t = 0.01$, the correct evolution of the drag coefficient is not well predicted with the Euler scheme (not

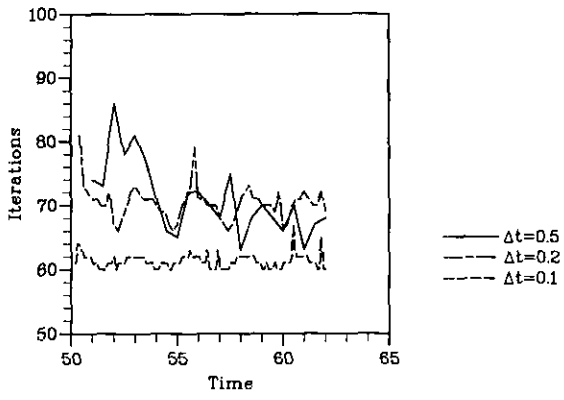

FIG. 36. Backward-differencing.

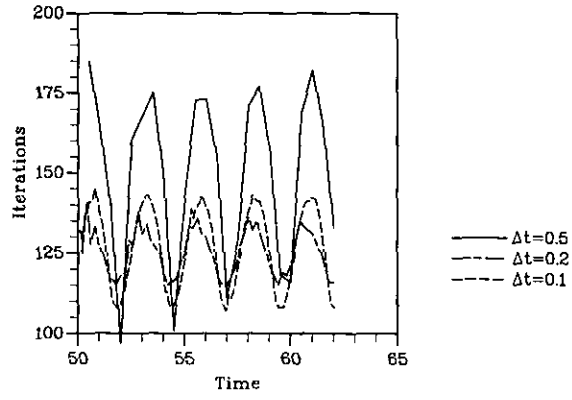

FIG. 38. DIRK2.1.

shown). Comparing the time evolution of the streamwise velocity in the wake, Figs. 31, 33–35, it is found that the most accurate scheme is the DIRK2.1 scheme. For $\Delta t = 0.2$, the DIRK2.1 and DIRK2.2 schemes give identical solutions. It can be seen that with a time step of $\Delta t = 0.5$, the solution computed with the DIRK2.2 scheme departs more from the reference solution than when it is calculated with the DIRK2.1 scheme. A further degradation of the accuracy is found with the Crank–Nicolson scheme. From Fig. 34, it appears that the solution computed with the Crank–Nicolson scheme with $\Delta t = 0.5$ is less accurate than the solution computed with the DIRK2.1 scheme with $\Delta t = 0.8$. However, with $\Delta t = 0.2$, the Crank–Nicolson scheme predicts the correct signal. Such is not the case with the backward-differencing scheme. A noticeable deviation from the reference solution is observed with $\Delta t = 0.2$. Even with a time step of $\Delta t = 0.1$, the velocity evolution calculated with the backward-differencing scheme departs slightly from the reference solution. For the simulation of periodic motions, it thus appears that the non-dissipativity—for imaginary eigenvalues λ—of the integration scheme is important to enable the use of relatively large time steps. On the other hand, when non-dissipative schemes are employed, moderately large time steps can be employed without any degradation of the time signals. It may be noted that much larger time steps than the one reported in [3, 28] were employed in the present simulations. In agreement with the computation performed on the model problem, it was found that with a specially designed Runge–Kutta scheme approximately 16 time steps per period are sufficient for resolving accurately a periodic signal. In comparison, the backward-differency scheme requires nearly three times as many steps to reach a similar level of accuracy.

4.5.3. *Investigation on the Efficiency of Integration Schemes*

In Figs. 36–39, the evolution of the number of iterations necessary to solve the pseudo-steady system (4) is presented for the different integration schemes. The tolerance parameter ε was set to $\varepsilon = 10^{-4}$. The CPU times required for the above computations are listed in Table VI. Examining

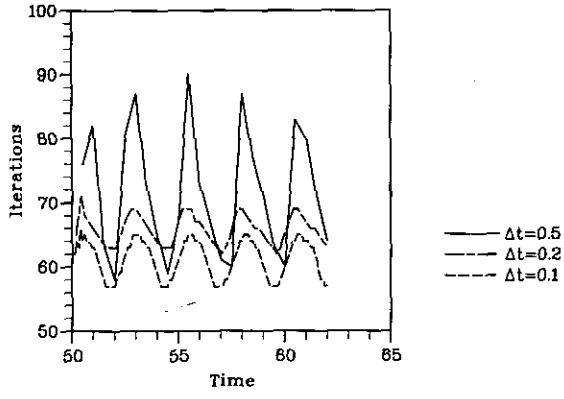

FIG. 37. Crank–Nicholson.

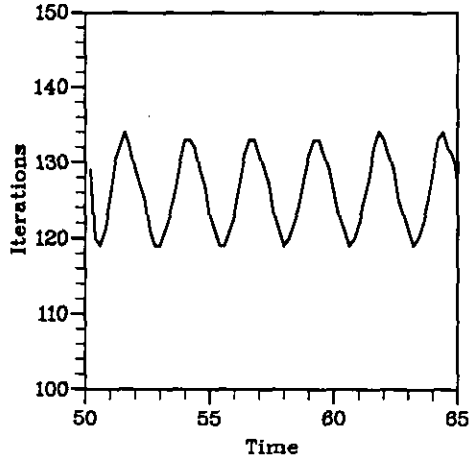

FIG. 39. Runge–Kutta (2), 100×60 mesh, $\Delta t = 0.2$.

TABLE VI

Execution Characteristics on the Vortex Shedding Problem, Re = 200

Scheme	Δt	Pseudo CFL number	Run CPU time (s)
Backward-differencing	0.1	25	430
	0.2	20	247
	0.5	10	106
Crank–Nicolson	0.1	25	431
	0.2	20	232
	0.5	10	101
DIRK 2.1	0.1	30	893
	0.2	25	441
	0.5	15	195
DIRK 2.2	0.2	25	441
	0.5	15	205

Figs. 36–39 and Table VI, it can be observed that the integration schemes behave differently for the vortex shedding problem than for the impulsively started cylinder problem. The strongly A-stable characteristic of the scheme, that was shown to be extremely important in the simulation of the impulsively started cylinder problem, appears to have no influence on the convergence properties of the scheme for the periodic problem. For instance, it can be seen that similar CPU times are required with the backward-differencing and the Crank–Nicolson schemes and with the two Runge–Kutta schemes. From Figs. 36–39, the periodicity of the signal is clearly visible in the convergence histories. Only the backward-differencing scheme with $\Delta t = 0.5$ shows a tendency of decreasing iteration numbers with time. On closer examination, it is found that with this large value of Δt, the computed solution differs significantly from the reference solution. Thus, for the backward-differencing scheme with $\Delta t = 0.5$, the initial field is not a solution of the periodic field at a given time. Therefore, a transient phase develops before the establishment of a (wrong) periodic phase.

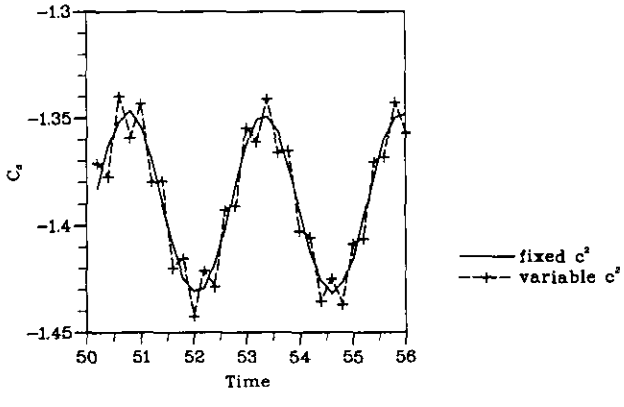

FIG. 40. Crank–Nicholson, 100×60 mesh, $\Delta t = 0.2$.

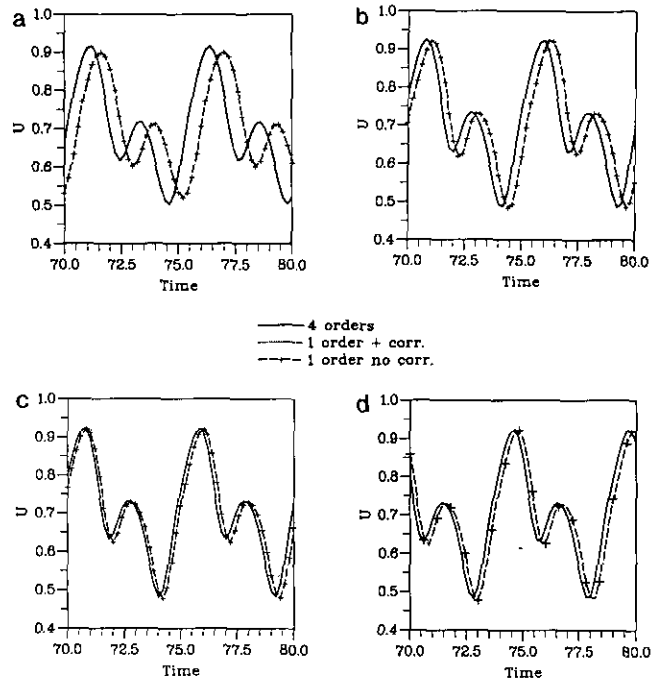

FIG. 41. (a), (c) Backward-differencing, 100×60 mesh, $\Delta t = 0.2$. (b), (d) Crank–Nicholson, 100×60 mesh, $\Delta t = 0.2$.

It can be observed that for all the schemes, the average number of iterations necessary to solve system (4) is nearly independent of the time step Δt. As for the impulsively started cylinder problem, the simulations were conducted using a nearly optimal CFL number for each case. It has been found during this optimization step, that with a CFL number larger than the optimal CFL number the amplitude of the oscillations visible in Figs. 36–39 is reduced, but the

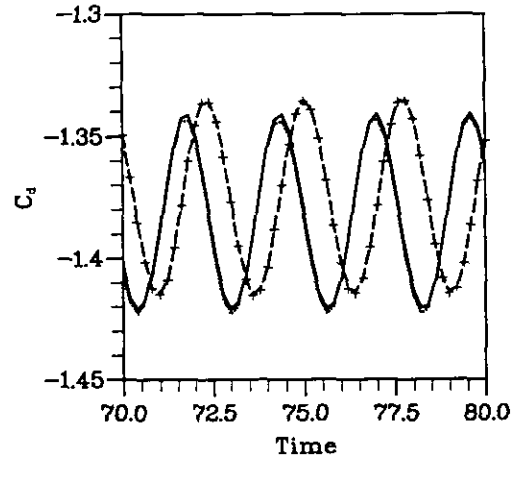

FIG. 42. Backward-differencing, 100×60 mesh, $\Delta t = 0.2$.

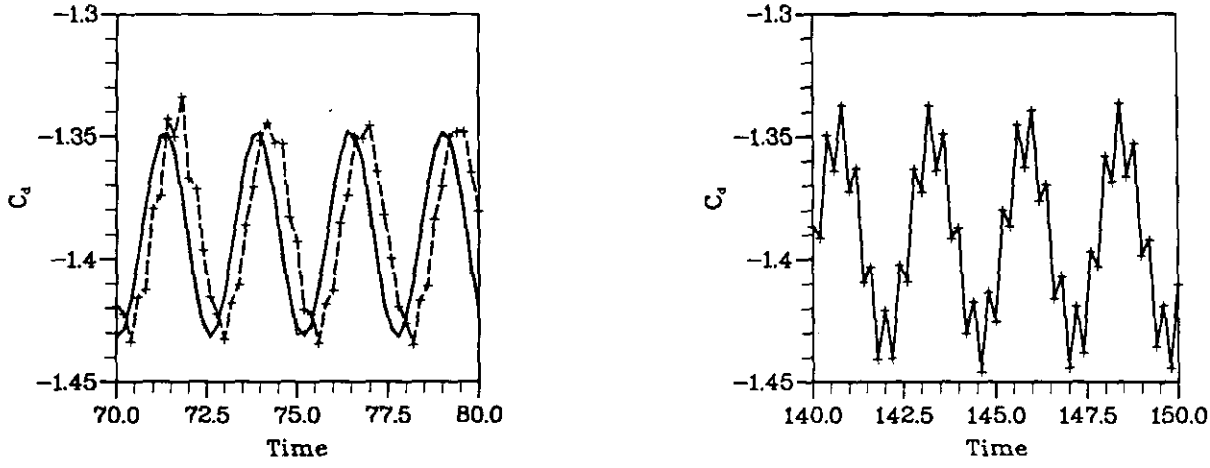

FIG. 43. Crank–Nicholson, 100×60 mesh, $\Delta t = 0.2$.

average number of iterations increases. From Table VI it can also be seen that the optimal CFL numbers are slightly larger with smaller time steps. However, the influence of the time step on the optimal CFL number is marginal. This weak dependence is probably the consequence of the lack of time derivative in the continuity equation since the factorization error is scaled by

$$\left(I + \sigma \frac{\Delta\tau}{\Delta t}\tilde{I}\right)^{-2} \frac{\Delta\tau}{\Delta x} A \frac{\Delta\tau}{\Delta y} B. \tag{14}$$

If in Eq. (14) the identity matrix appeared in place of $\tilde{I}$, the factorization error would be scaled, for sufficiently large $\Delta\tau/\Delta t$, by

$$\frac{1}{\sigma^2}\left(\frac{\Delta tA}{\Delta x}\frac{\Delta tB}{\Delta y}\right).$$

Thus, the factorization error would not depend any more on the pseudo-CFL number, and large pseudo-CFL numbers could probably be used.

Finally, when the efficiency of the different schemes is compared, it is found that the Crank–Nicolson scheme requires the least computing time for a given accuracy. The Runge–Kutta and backward-differencing schemes appear to have similar efficiencies. However, it should be mentioned that the results obtained with the Crank–Nicolson scheme were computed with a fixed artificial compressibility coefficient c^2. The reason is clearly visible in Fig. 40; oscillations are found on the drag coefficient when the parameter c^2 is computed using Eq. (6) (c^2 varying between $c^2 = 5.95$ and $c^2 = 6.55$). The Crank–Nicolson scheme appears thus extremely sensitive, since fixed c^2 should be used for the vortex shedding problem, whereas variable c^2 must be employed for the impulsively started cylinder problem. This sensitivity is further confirmed by studying the influence of

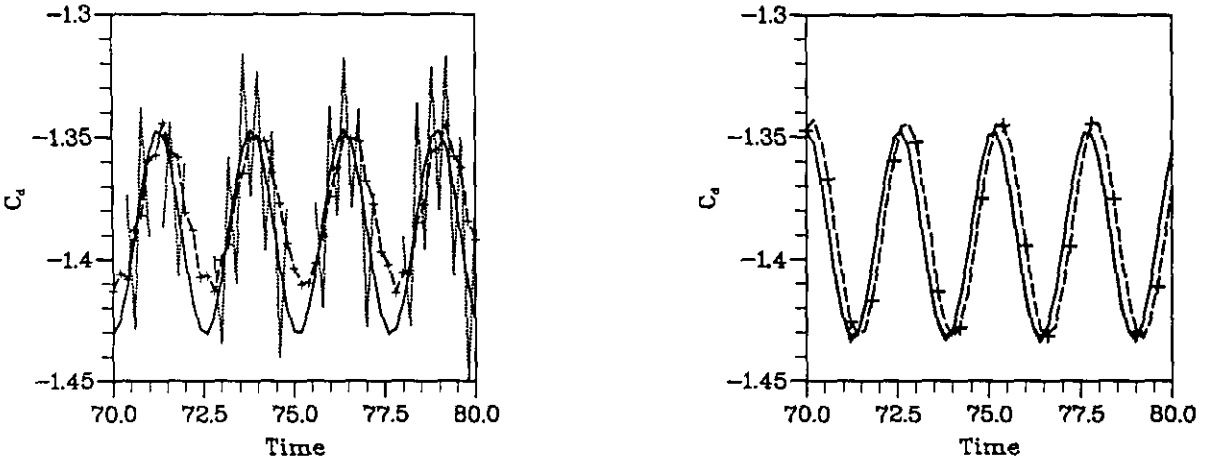

FIG. 44. Runge–Kutta, 100×60 mesh, $\Delta t = 0.2$.

FIG. 45. Runge–Kutta (2), 100×60 mesh, $\Delta t = 0.2$.

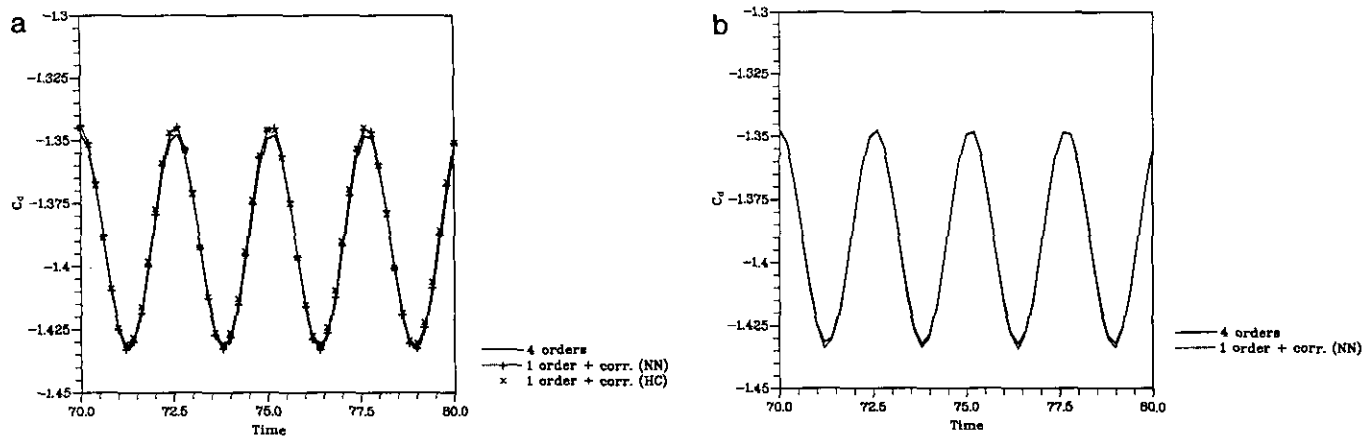

FIGURE 46

an approximate solution of the pseudo-steady system on the accuracy of the method.

4.5.4. *Influence of the Convergence Criterion*

For the different integration schemes described in Section 2, the influence of the tolerance level—used to solve the pseudo-steady system (4)—on the accuracy of the simulations has been studied. Starting from the same initial field at $t = 50$, two series of computations were performed. In the first series, the simulations were conducted with the different schemes using a tight tolerance level, $\varepsilon = 10^{-4}$. In a second series, the simulations were repeated using the less stringent value $\varepsilon = 10^{-1}$. Figures 41a–d illustrate the influence of the tolerance level ε on the evolution of the streamwise velocity "measured" in the wake, 2.5 diameters behind the cylinder. It can be observed that the time correction techniques are, in this case, very efficient, since for all the integration schemes identical signals are found with $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-1}$. Without correction, the variation of the velocity signal is well reproduced but the frequency is underestimated. A correlation between the frequency error, introduced by the approximate solution of the pseudo-steady system, and the accuracy of the integration schemes is also observed; the higher the accuracy, the smaller the solution error, ε^{n+1}.

Whereas the procedure described in Section 2.2 appears to be efficient in reproducing the velocity signal in the wake, for the Crank–Nicolson and the DIRK2.1 schemes the correction procedure does not improve the time evolution of the drag coefficient, Figs. 43, 44. With the correction step, the dominant frequency is correctly predicted, but spurious high frequencies are added to the signal. These high frequencies can be quite large, Fig. 44. The high frequencies are gradually created by the DIRK2.1 scheme. With the Crank–Nicolson scheme, the generation of these spurious frequencies is more erratic. They are generated immediately, and after some time they almost disappear (between $t \simeq 50$ and $t \simeq 90$), Fig. 43a, but finally a spurious periodic signal establishes itself, Fig. 43b. With the backward-differencing and the DIRK2.2 schemes, either correction procedure seems appropriate, Figs. 42, 45. Again, it is found that non-strongly A-stable schemes are sensitive. As for the impulsively started cylinder problem, the form of the correction procedure used with the DIRK2.1 scheme appears to affect significantly the efficiency of the correction step stage. In Fig. 46a, it can be seen that the correction procedure can be efficient with the DIRK2.1 scheme, if the correction is performed only after the second stage. However, comparing Figs. 46a and b, it is found that the correction procedure is even more efficient with the DIRK2.2 scheme; the solutions calculated with $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-1}$ differ only in the vicinity of the minima.

In contrast to the results obtained on the impulsively started cylinder problem, a significant reduction in CPU time is achieved when a large tolerance parameter, $\varepsilon = 10^{-1}$, is used in combination with the "correction (N)" procedure, Table VII. The correction procedure seems, in this case, to

TABLE VII

Influence of the Precision Used to Solve the Pseudo-steady System

Scheme	ε	Run CPU time (s)	Correction
DIRK2.1	10^{-4}	441	—
	10^{-1}	215	corr (O)
	10^{-1}	131	corr (HC)
	10^{-1}	103	corr (NN)
	10^{-1}	76	no corr
DIRK2.2	10^{-4}	441	—
	10^{-1}	94	corr (NN)
	10^{-1}	69	no corr

Note. Vortex shedding problem, Re = 200.

affect the accuracy of the results directly, and not only because of the increase in the number of iterations necessary to solve the pseudo-steady system. As in the previous investigations, it is found that the "correction (N)" procedure must be always preferred to the "correction (O)" procedure.

5. CONCLUSIONS

A numerical procedure for the simulation of unsteady incompressible flows has been presented. The method consists of combining an A-stable time discretization with the artificial compressibility method. The latter method is used to solve, at each time step, the non-linear system arising from the time discretization. It has been shown that, with this procedure, the time step used in the simulations is only limited by accuracy considerations; thus, large time steps can be used to accurately simulate flows varying slowly.

Several A-stable integration schemes were investigated and tested under different flow conditions. From these numerical experiments, it was observed that the solution of the non-linear system requires almost as many iterations with small and large time steps. Therefore, the numerical technique employed in this work may not be well suited for the simulation of flows requiring very small time steps, e.g., direct simulation of turbulence. The various calculations have shown that the integration schemes behave differently for unsteady flows driven by fast transient motions and for periodic flows. For the former, it was found that the use of *strongly* A-stable integration schemes is mandatory. Strongly A-stable schemes lead then to efficient and robust solvers. Otherwise, like with the Crank–Nicolson scheme, the computations are very sensitive to the type of flow-field. For the simulation of periodic flows, strong A-stability of the scheme was shown to be less important. For this class of flows, it was found more useful to employ non-dissipative schemes since larger time steps can be used without affecting significantly the accuracy of the solution. The strong A-stable property was, however, shown to reduce, for these flow regimes also, the sensitivity of the scheme. For instance, it was observed that, when strongly A-stable schemes are combined with a time correction procedure, good solutions can be obtained even if, at each iteration, the non-linear system is not solved accurately. Several time correction procedures were tested. The best results were obtained when using a procedure derived from Hirt and Harlow [18]. From the various simulations performed, it appears that, among the tested schemes, the best compromise in accuracy, efficiency, and robustness is the DIRK 2.2 scheme. With this scheme, the technique used for the simulation of unsteady flows is very promising, since the method appears to be robust and since solutions acceptable for most engineering applications can be computed at reduced expenses by combining large time steps with an approximate solution of the non-linear systems.

APPENDIX: EXPRESSIONS OF THE SPLIT JACOBIAN MATRICES $A_\eta^\pm$

$$A_\eta^+ = \frac{1}{2}\begin{bmatrix} c^2 & c^2\lambda^+ n_x & c^2\lambda^+ n_y & c^2\lambda^+ n_z \\ \lambda^+ n_x + \Omega_x\alpha^+ & (\lambda^{+2} - 2a\check{u}^+)\, n_x^2 + \Omega_x\beta^+ n_x + 2a\check{u}^+ & (\lambda^{+2} - 2a\check{u}^+)\, n_x n_y + \Omega_x\beta^+ n_y & (\lambda^{+2} - 2a\check{u}^+)\, n_x n_z + \Omega_x\beta^+ n_z \\ \end{bmatrix}$$

ACKNOWLEDGMENTS

This work has been supported by a joint CERS-Sulzer Brothers S.A. contract and the computing time was provided by the Service Informatique Central of EPFL. The author also thanks Professor I. L. Ryhming for his support throughout the course of this study.

REFERENCES

1. M. Breuer and D. Hänel, In *8th GAMM Conference on Numerical Methods in Fluid Mechanics*, edited by P. Wesseling (Vieweg, Brunswick, 1989), p. 42.
2. S. E. Rogers and D. Kwak, *AIAA J.* **28**(2), 253 (1990).
3. M. Rosenfeld, D. Kwak, and M. Vinokur, *J. Comput. Phys.* **94**, 102 (1991).
4. S.-W. Kim and T. J. Benson, *Comput. Fluids* **21**, 435 (1992).
5. S. V. Patankar and D. B. Spalding, *Int. J. Heat Mass Transfer* **15**, 1787 (1972).
6. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC, 1980).
7. J. P. Van Doormaal and G. D. Raithby, *Numer. Heat Mass Transfer* **7**, 147 (1984).
8. R. I. Issa, *J. Comput. Phys.* **62**, 40 (1985).
9. A. J. Chorin, *Math. Comput.* **22**, 745 (1968).
10. J. Kim and P. Moin, *J. Comput. Phys.* **59**, 308 (1985).
11. A. J. Chorin, *J. Comput. Phys.* **2**, 12 (1967).
12. J. L. C. Chang and D. Kwak, *AIAA Paper* 84-0252, 1984.
13. F. H. Harlow and J. E. Welch, *Phys. Fluids* **8**, 2182 (1965).
14. J. L. Steger and P. Kutler, *AIAA J.* **15**(4), 581 (1977).
15. R. Peyret, *J. Fluid Mech.* **78**, 49 (1976).
16. A. Rannacher, Sonderforsungsbereich 123,492, Universität Heidelberg, 1988.
17. E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations I, Nonstiff problems*, Springer Series in Computational Mathematics, Vol. 8 (Springer-Verlag, New York/Berlin, 1987).
18. C. W. Hirt and F. H. Harlow, *J. Comput. Phys.* **2**, 114 (1967).
19. Y. P. Marx, *Int. J. Numer. Methods Fluids* **13**(6), 775 (1991).
20. E. Turkel, *J. Comput. Phys.* **72**, 277 (1987).
21. Y. P. Marx, Technical Report T-92-2, IMHEF, 1992 (unpublished).
22. P. L. Roe, *J. Comput. Phys.* **43**, 357 (1981).
23. Y. P. Marx, *Commun. Appl. Numer. Methods* **8**, 633 (1992).
24. D. S. Chaussee and T. H. Pulliam, *AIAA Paper* 80-0067, 1980.
25. R. Bouard and M. Coutanceau, *J. Fluid Mech.* **101**, 583 (1980).
26. Y. Lecointe and J. Piquet, *von Kármán Lecture Ser.*, 1985-04, 1985.
27. W. M. Collins and S. C. R. Dennis, *J. Fluid Mech.* **60**, 105 (1973).
28. Y. Lecointe and J. Piquet, *Comput. Fluids* **12**, 255 (1984).
29. S. S. Abarbanel, W. S. Don, D. Gottlieb, D. H. Rudy, and J. C. Townsend, *J. Fluid Mech.* **225**, 557 (1991).
30. R. Wille, *Adv. Appl. Mech.* **6** (1960).
31. J. H. Gerrard, *Philos. Trans. R. Soc.* **288** (1978).