INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **29**: 35–57 (1999)

SOLUTION OF INCOMPRESSIBLE FLOWS WITH OR WITHOUT A FREE SURFACE USING THE FINITE VOLUME METHOD ON UNSTRUCTURED TRIANGULAR MESHES

C.T. CHAN AND K. ANASTASIOU*

Department of Civil Engineering, Imperial College of Science, Technology and Medicine, Imperial College Road, London SW7 2BW, UK

SUMMARY

An incompressible Navier–Stokes solver based on a cell-centre finite volume formulation for unstructured triangular meshes is developed and tested. The solution methodology makes use of pseudocompressibility, whereby the convective terms are computed using a Godunov-type second-order upwind finite volume formulation. The evolution of the solution in time is obtained by subiterating the equations in pseudotime for each physical time step, with the pseudotime step set equal to infinity. For flows with a free surface the computational mesh is fitted to the free surface boundary at each time step, with the free surface elevation satisfying a kinematic boundary condition. A 'leakage coefficient', ε, is introduced for the calculation of flows with a free surface in order to control the leakage of flow through the free surface. This allows the assumption of stationarity of mesh points to be made during the course of pseudotime iteration. The solver is tested by comparing the output with a wide range of documented published results, both for flows with and without a free surface. The presented results show that the solver is robust. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: Navier–Stokes solver; finite volume method; unstructured triangular mesh; method of pseudocompressibility; free surface flows; leakage coefficient

1. INTRODUCTION

In recent years algorithms for solving the Euler and Navier–Stokes equations for compressible flows using unstructured meshes have gained in popularity and impressive results have been reported [1–5]. By exploiting the high flexibility of unstructured meshes and by adopting a solution methodology based on the finite volume method, the governing systems of equations can be solved on a Cartesian co-ordinate system without having to use co-ordinate transformations and, moreover, discontinuous flows like those associated with shock propagation can be successfully simulated. Many researchers have extended techniques invented for compressible flows into the domain of incompressible flows. One of these extensions involves the application of Godunov-type second-order upwind finite volume methods on triangular meshes for the solution of the system of the shallow water equations [6]. Another, perhaps the most notable one, is the introduction of pseudocompressiblility into the system of incompressible Navier–

* Correspondence to: Department of Civil Engineering, Imperial College of Science, Technology and Medicine, Imperial College Road, London SW7 2BW, UK.

CCC 0271–2091/99/010035–23$17.50

Received January 1997
Revised March 1997

Stokes equations and solving the resulting set of equations using techniques originally developed for compressible flow problems [7–15].

In the present work the method of pseudocompressibility is adopted for the solution of the incompressible Navier–Stokes equations on unstructured triangular meshes. The merit of using pseudocompressibility is that it directly couples the pressure and velocity fields at the same time level thus producing a hyperbolic system of equations. As a result, some of the recently developed upwind finite volume solution techniques, such as the characteristics based Godunov-type schemes, can be applied in a straightforward manner to incompressible flows. The main disadvantage of introducing pseudocompressibility into the system of incompressible Navier–Stokes equations is that it is not suitable for time-dependent solutions and, as a result, only steady state solutions can be obtained. However, this undesirable aspect has been overcome by the emergence of the implicit pseudotime iteration technique [11–15], whereby the time-dependent solution is viewed as the solution of a steady state problem at each time step. In applying this technique, the pseudotime is usually set to a very large value which leads to fast convergence. In the present study the pseudotime is set to infinity. As a result, accurate time-dependent solutions are shown to be obtainable by the method of pseudocompressibility solely by implicit iteration.

When applied to unstructured meshes the above solution methodology is also suitable for problems involving flows with a free surface. Hino *et al.* [16] have recently reported the use of a similar solution methodology for steady state problems. For flows with a free surface the computational mesh is adapted to the free surface at the end of each computational time step, with the free surface elevation satisfying a kinematic boundary condition. The time history of the mesh evolution is not critical for a steady state solution since the mesh becomes stationary at steady state. As a result, the time derivative of the computational cell area is eliminated from the system of discretized algebraic equations. However, this is not the case for time-dependent calculations since this derivative is not identically equal to zero during the transient stage. This poses a problem to the solution technique when applied to unstructured triangular meshes, as it is by no means easy to quantify the movement of the triangular cells. In the present work this difficulty is dealt with by discarding the time derivative of the computational cell area in the discretized equations and, consequently, the mesh is assumed to be stationary during the course of pseudotime iteration. In order to compensate for any side effects arising from this simplification, a 'leakage coefficient' ε, is introduced which allows for some 'leakage' of flow through the free surface during the implicit iteration stage. Results obtained show that this technique is viable and robust.

The present numerical model is second-order-accurate and is based on a cell-centre finite volume upwind scheme. The description of the flow variables over each triangular cell is based upon the technique proposed by Barth and Jespersen [4]. Roe's flux function is used for the evaluation of the inviscid fluxes at the triangular cell faces, assuming a local Riemann problem in the direction normal to a cell face. The viscous terms are computed in a manner similar to that proposed by Pan and Cheng [5] which is second-order-accurate.

Section 3 of the paper describes the finite volume discretization of the 2D Navier–Stokes equations on an unstructured triangular mesh. A detailed account of the formulation of the inviscid and viscous fluxes is given in Section 4. In Section 5 the implementation of appropriate boundary conditions is presented, and the procedures used for updating the free surface boundary as well as for applying the leakage coefficient are discussed. The integration in time of the discretized equations is described in Section 6. Results for internal flows such as flow over an oscillating plate, flow through a one-dimensional channel with an oscillating back pressure, flow over a circular cylinder, and flow over a backward-facing step are presented and

discussed in Section 7. In order to fully validate the solution methodology results are also presented in Section 7 for flows with a free surface, such as the oscillation of the free surface of the water in a tank, a water column subjected to a centrifugal force, propagation of a solitary wave, and flow over a submerged hydrofoil. The presented results show that the present scheme is accurate and robust.

2. FINITE VOLUME DISCRETIZATION OF THE 2D NAVIER–STOKES EQUATIONS ON AN UNSTRUCTURED MESH

The integral form of the system of non-dimensionalized 2D Navier–Stokes equations can be written as

$$\frac{\partial}{\partial t}\iint_{\Omega} Q \, \mathrm{d}\Omega + \int_{S} \mathbf{F}\cdot\mathbf{n} \, \mathrm{d}S = \iint_{\Omega} H \, \mathrm{d}\Omega, \tag{1}$$

where Ω is the domain of interest, S is the boundary surrounding Ω, $\mathbf{n}$ is the unit normal to S in the outward direction, Q is the vector of primitive variables, $\mathbf{F}$ is the vector of flux functions through S, and H is the body forcing function. Q, $\mathbf{F}$ and H are represented by the following expressions:

$$\mathbf{F} = F^{\mathrm{I}} - F^{\mathrm{V}} = \left(f^{\mathrm{I}} - \frac{1}{Re} f^{\mathrm{V}}\right) n_x + \left(g^{\mathrm{I}} - \frac{1}{Re} g^{\mathrm{V}}\right) n_y, \qquad Q = \begin{bmatrix} 0 \\ u \\ v \end{bmatrix}, \qquad f^{\mathrm{I}} = \begin{bmatrix} u \\ u^2 + p \\ uv \end{bmatrix},$$

$$g^{\mathrm{I}} = \begin{bmatrix} v \\ uv \\ v^2 + p \end{bmatrix}, \qquad f^{\mathrm{V}} = \begin{bmatrix} 0 \\ u_x \\ v_x \end{bmatrix}, \qquad g^{\mathrm{V}} = \begin{bmatrix} 0 \\ u_y \\ v_y \end{bmatrix}, \qquad H = \begin{bmatrix} 0 \\ -s_x/F^2 \\ 0 \end{bmatrix}, \tag{2}$$

where the superscripts I and V denote the inviscid and viscous fluxes, respectively, n_x and n_y denote the components of the normal vector $\mathbf{n}$, p is the dynamic pressure, u and v are the velocity components, u_x, u_y and v_x, v_y, are the derivatives of the velocity components in the x- and y-directions, respectively, s_x is the free surface slope, Re is the Reynolds number, and F is the Froude number. It should be noted that for free surface flow the pressure p would correspond to the kinematic pressure, which is equal to the dynamic pressure minus the hydrostatic pressure.

In order to couple the pressure and velocity fields at the same time level, so as to produce a hyperbolic system of equations, the pseudocompressibility relation is introduced by adding the time derivative of pressure to the continuity equation. Accordingly, Q, f^{I} and g^{I} are modified as follows:

$$Q = \begin{bmatrix} p \\ u \\ v \end{bmatrix}, \qquad f^{\mathrm{I}} = \begin{bmatrix} \beta u \\ u^2 + p \\ uv \end{bmatrix}, \qquad g^{\mathrm{I}} = \begin{bmatrix} \beta v \\ uv \\ v^2 + p \end{bmatrix}, \tag{3}$$

where β is the coefficient of pseudocompressibility. It is clear from the above expressions that Q, f^{I} and g^{I} of Equation (2) are recovered from Equation (3) only when steady state is reached. Therefore, and in order to obtain accurate time-dependent solutions, the above formulation needs further modification which is discussed in Section 6.

The spatial discretization of Equation (1) is carried out on an unstructured computational domain formed by a collection of triangular cells. The average of conserved variables is stored at the centre of each cell, and the edges of each cell define the faces of a triangular control volume. For each triangular control volume, Equation (1) is written as

$$\frac{\partial Q_i V_i}{\partial t} = -\oint_{\partial C_i} \mathbf{F}\cdot\mathbf{n}\,\mathrm{d}S + H_i V_i = -R(Q_i), \tag{4}$$

where Q_i and H_i are the average quantities of cell i stored at the cell center, and ∂C_i and V_i denote the boundary of the cell and the area of cell i, respectively. The right-hand side of Equation (3) is evaluated by summing the flux vectors over each edge of the triangular cell, and the discrete form of the integral is

$$\oint_{\partial C_i} \mathbf{F}\cdot\mathbf{n}\,\mathrm{d}S = \sum_{j=k(i)} F_{i,j}\Delta l_j, \tag{5}$$

where $k(i)$ is a list of the neighbouring cells to cell i, $F_{i,j}$ is the numerical flux through the interface of cells i and j, and Δl_j is the length of edge j. For a cell-centered scheme based on a triangular mesh $k(i)$ is a list of three cells.

3. EVALUATION OF INVISCID AND VISCOUS FLUXES

3.1. Inviscid fluxes

In order to evaluate the inviscid numerical fluxes $F^{\mathrm{I}}_{i,j}$, Roe's [17–19] flux function is adopted locally at each cell edge, assuming a 1D Riemann problem in the direction normal to the cell edge, as follows:

$$F^{\mathrm{I}}_{i,j} = \frac{1}{2}[F^{\mathrm{I}}(Q^+_{i,j}) + F^{\mathrm{I}}(Q^-_{i,j}) - |A|(Q^+_{i,j} - Q^-_{i,j})], \qquad |A| = R|\Lambda|L, \tag{6}$$

where $Q^+_{i,j}$ and $Q^-_{i,j}$ are the reconstructed right and left states of the cell face between cell i and cell j, A is the flux Jacobian evaluated by Roe's average state, which is the average of Q^+ and Q^- [20]. The quantities R, L and Λ are the right and left eigenvectors of A and the eigenvalues of A, respectively. The flux Jacobian A is

$$A = \frac{\partial(\mathbf{F}\cdot\mathbf{n})}{\partial Q} = \begin{bmatrix} 0 & \beta n_x & \beta n_y \\ n_x & 2un_x + vn_y & un_y \\ n_y & vn_x & un_x + 2vn_y \end{bmatrix}, \tag{7]}$$

and its eigenvalues are given by

$$\lambda_1 = un_x + vn_y, \qquad \lambda_2 = un_x + vn_y + c, \qquad \lambda_3 = un_x + vn_y - c. \tag{8}$$

The right eigenvector is

$$\begin{bmatrix} 0 & \beta c & -\beta c \\ n_y & u\lambda_2 + \beta n_x & u\lambda_3 + \beta n_x \\ -n_x & v\lambda_2 + \beta n_y & v\lambda_3 + \beta n_y \end{bmatrix}, \tag{9}$$

and the left eigenvector matrix is

$$\frac{1}{c^2}\begin{bmatrix} -(un_y - vn_x) & v\lambda_1 + \beta n_y & -(u\lambda_1 + \beta n_x) \\ -\lambda_3/(2\beta) & n_x/2 & n_y/2 \\ -\lambda_2/(2\beta) & n_x/2 & n_y/2 \end{bmatrix}, \tag{10}$$

where c in the above expressions denotes the pseudowave celerity $[(un_x + vn_y)^2 + \beta(n_x^2 + n_y^2)]^{1/2}$, and n denotes the magnitude of the normal vector $\mathbf{n}$ to the cell edge, for a unit normal the magnitude is equal to 1.

In order to obtain second-order accuracy, a piecewise linear model for the cell variables must first be reconstructed from the solution before the two Riemann states at each cell edge are computed. For a given cell with centre A for example, this requires the construction of the cell variables in the form

$$Q(x, y) = Q_A + \nabla Q_A \cdot \mathbf{r}, \tag{11}$$

where $\mathbf{r}$ is the vector from the cell centre A to any point (x, y) within the cell, Q_A is the cell centre value of cell A, and ∇Q_A is the gradient of cell A evaluated using the cell centre values.

By applying Gauss's theorem, the solution gradient of cell A can be estimated by evaluating a boundary integral over the path surrounding A as

$$\nabla Q = \frac{1}{A_\Omega} \oint_{\partial A} Q\mathbf{n}\, \mathrm{d}S, \tag{12}$$

where ∂A is an integration path connecting points with known variables, and A_Ω is the area contained in the path. A direct simple integration path for ∇Q is the path connecting three neighbouring cell centres B, C and D as shown in Figure 1. As suggested by Pan and Cheng [5], the estimated gradient accuracy can be further improved by carrying out a weighted averaging process, as follows:

$$\nabla Q_A^{\mathrm{mod}} = \frac{1}{2}(\nabla Q_A + w_B \nabla Q_B + w_C \nabla Q_C + w_D \nabla Q_D), \tag{13}$$

and

$$w_B = (\mathbf{r}_{AC} \times \mathbf{r}_{CD})/(\mathbf{r}_{BC} \times \mathbf{r}_{CD}), \qquad w_C = (\mathbf{r}_{AD} \times \mathbf{r}_{DB})/(\mathbf{r}_{BC} \times \mathbf{r}_{CD}),$$
$$w_D = (\mathbf{r}_{AB} \times \mathbf{r}_{BC})/(\mathbf{r}_{BC} \times \mathbf{r}_{CD}), \tag{14}$$

where $\nabla Q_{B,C,D}$ are the gradient vectors of cells B, C, D, respectively, evaluated by Equation (11), and the quantities $\mathbf{r}$ are the distance vectors from cell centre A to the centres of the neighbouring cells.

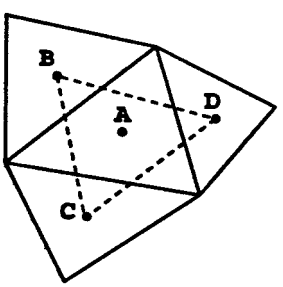

Figure 1. *BCD* defines the integration path for ∇Q_A.

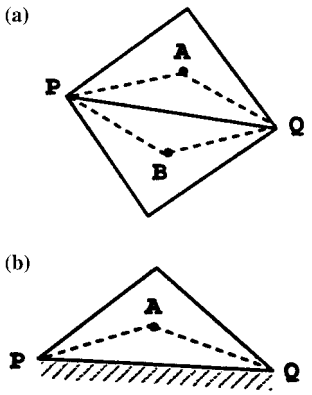

Figure 2. (a) *APBQ* defines the integration path for viscous stresses at an interior cell; (b) APQ defines the integration path for viscous stresses at a boundary cell.

3.2. Viscous fluxes

The viscous numerical flux $F^{\mathrm{V}}_{i,j}$ is evaluated by computing the gradient of the velocity at each cell edge using the expression

$$\nabla Q_{PQ} = \frac{1}{A_{APBQ}} (A_{BPQ} \nabla Q_{APQ} + A_{APQ} \nabla Q_{BPQ}), \tag{15}$$

where, with reference to Figure 2(a), ∇Q_{APQ} and ∇Q_{BPQ} are the gradient vectors of paths APQ and BPQ, respectively, evaluated by Equation (17), and A_{APBQ}, A_{APQ}, and A_{BPQ} are the areas of quadrilateral $APBQ$ and triangles APQ and BPQ, respectively. The variable values at cell vertices P and Q are obtained by a distance-weighted averaging of cell centre values surrounding that particular vertex. Once the gradient vectors of velocity are obtained, the viscous fluxes at cell edge PQ can then be combined with the evaluated inviscid flux and put into Equation (4) for integration. For a triangle on the boundary as shown in Figure 2(b), the gradient at the boundary PQ is evaluated on the triangular contour APQ only. Assuming accurate values at P and Q, the above formulation is second-order-accurate for a parallelogram and first-order-accurate for a general quadrilateral element [5].

4. BOUNDARY CONDITIONS AND FREE SURFACE TREATMENT

4.1. Boundary conditions

At inflow or outflow boundaries, characteristic boundary conditions are imposed. Following this treatment, characteristic variables are extrapolated from interior of the domain whenever applicable. For example, with reference to the local cell edge axes, there always exists one outgoing characteristic at the inflow boundary and two outgoing characteristics at the outflow boundary. The extrapolation is carried out linearly using the left eigenvector matrix as shown below:

$$\begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_{\mathrm{B}} = \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_{\mathrm{I}}, \tag{16}$$

where the symbol B above denotes the boundary values or the values of the right Riemann states at the boundary, and I denotes the imposed values. Imposed values are values either imposed physically or given by the values of the left Riemann states at the boundary. The vector $[p, u, v]^{-1}$ in Equation (16) may also be replaced by $[\Delta p, \Delta u, \Delta v]^{-1}$. Boundary conditions are applied at the interface of a cell.

The boundary conditions for a wall boundary are set as

Slip wall:	No − slip wall:
$h_{\mathrm{B}} = h_{\mathrm{I}}$	$h_{\mathrm{B}} = h_{\mathrm{I}}$
$u_{\mathrm{B}} = 0$	$u_{\mathrm{B}} = 0$
$v_{\mathrm{B}} = v_{\mathrm{I}}$	$v_{\mathrm{B}} = 0$

(17)

4.2. Free surface treatment

Two free surface boundary conditions must be satisfied. The first is the dynamic condition which states that stresses on the air–liquid interface are continuous. In the present study the tangential stress and surface tension are set equal to zero at the air–liquid interface, and the air pressure is assumed to be constant. The second free surface condition is the kinematic condition which states that the free surface is a material surface. This means that a fluid particle on the free surface remains there. The dynamic and kinematic boundary conditions are expressed as

$$p = \text{constant}, \tag{18}$$

and

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} - v = 0, \tag{19}$$

where h is the free surface elevation with respect to the still water level. Equation (19) permits solutions only when h is single-valued.

The free surface boundary is calculated by solving Equation (19) at each time step following the convergence of the bulk flow. In order to accomplish this, the velocity components u and v on each free surface node are first extrapolated from the interior of the domain using the calculated values from the surrounding cells. Equation (19), which is discretized spatially in the same manner as for the bulk flow (by applying the cell centre upwind finite volume method in one dimension) is then integrated in time using the explicit fourth-order Runge–Kutta method.

Following the updating of the free surface, the computational mesh is moved to adapt to the free surface boundary. This is accomplished by stretching the mesh vertically using the relationship

$$x_i^{n+1} = x_i^n, \qquad y_i^{n+1} = \begin{cases} y_b + \dfrac{y_s^{n+1}(x_i) - y_b}{y_s^n(x_i) - y_b}\,(y_i^n - y_b) & \text{if } y_i^n > y_b, \\ y_i^n & \text{otherwise} \end{cases} \tag{20}$$

where y_b is a reference elevation above which mesh points are stretched, while mesh points below it remain stationary. In the above expression $y_s(x_i)$ is the free surface elevation at $x = x_i$ and is computed by linear interpolation when x_i does not coincide with the x-co-ordinate of the free surface nodes. This procedure produces a linear stretching of the mesh points between the free surface elevation y_s and the reference elevation y_b.

It should be noted that, since the present free surface mesh adaptation method is based on a simple one-dimensional stretching technique, it cannot be used for the simulation of overturning or breaking waves where the free surface is not single valued.

4.2.1. Leakage coefficient. The free surface in the present scheme is considered to be moving discretely in time. Accordingly, it is assumed to be stationary within the duration of a physical time step and it moves only at the end of each time step. As a result, the time derivative of the computational cell area is dropped from the discretized algebraic equations. In order to compensate for any side effects arising from this simplification, the requirement of a material free surface is relaxed somewhat during the course of the numerical solution. This is done by considering the free surface to behave like a surface between the two extremes of a material surface and an open boundary on which the normal velocity gradient is zero. This approach provides a desirable degree of flexibility when it comes to specifying the normal components of velocity at the free surface. It is, therefore, possible to set the right Riemann state of the velocity component normal to the free surface to be zero, or equal to the value of the left Riemann state, or to a value somewhere in between these two. This is accomplished by introducing a 'leakage coefficient' ε, such that

$$v_n(R_r) = \varepsilon v_n(L_r), \tag{21}$$

where v_n denotes the velocity component normal to the free surface, and R_r and L_r denote the right and left Riemann states, respectively. When $\varepsilon = 0$, a true material free surface is recovered. When $\varepsilon = 1$, the free surface is set to behave like an open boundary on which the normal velocity gradient is zero.

It is worth highlighting that the leakage coefficient is introduced as an aid to attaining a time accurate solution for the evolving free surface. When the leakage coefficient is set equal to one, this amounts to simulating the free surface as a flexible membrane on which the pressure remains constant. When the leakage coefficient is set equal to zero, this amounts to setting the free surface as a rigid material surface, thus predefining its shape before the numerical computation starts. In this case the pressure of the left Riemann state will be different from that of the right Riemann state during the course of numerical iteration. Therefore, in general the leakage coefficient ε should be set equal to one in order to maintain time accuracy. If the setting of $\varepsilon = 1$ causes free surface instability, this is an indication that either the computational mesh at the free surface is too coarse or that the integration time step is too large. In such cases the coefficient can be used as a 'tuning' tool by assigning to it a value of < 1.

5. INTEGRATION IN TIME

Equation (4) can be integrated in time using the first-order Euler implicit time difference method which is expressed as

$$\frac{(QV)^{n+1} - (QV)^n}{\Delta t} = -R(Q^{n+1}), \tag{22}$$

where V is the computational cell area. For the purpose of accurate integration in time Equation (22) can not be used directly because a divergence free velocity field can not be recovered from the continuity equation due to the introduction of pseudocompressibility. As a result, Equation (22) must be further modified in order to attain a divergence free velocity field at each time step. This is accomplished by introducing a pseudotime derivative into the system of equations, as

$$\frac{(QV)^{n+1,m+1}-(QV)^{n+1,m}}{\Delta\tau}+I_{ta}\frac{(QV)^{n+1,m}-(QV)^{n}}{\Delta t}=-R(Q^{n+1,m+1}), \tag{23}$$

where τ is the pseudotime and $I_{ta}=\mathrm{diag}[0, 1, 1]$. The right-hand side (RHS) of Equation (23) can be linearized using Newton's method at the $m+1$ pseudotime level to yield

$$\left[I_m V+\frac{\partial R(Q^{n+1,m})}{\partial Q}\right](Q^{n+1,m+1}-Q^{n+1,m})=-\left[I_{ta}\frac{(Q^{n+1,m}-Q^{n})V}{\Delta t}+R(Q^{n+1,m})\right], \tag{24}$$

where $I_m=\mathrm{diag}[1/\Delta\tau, 1/\Delta\tau+1/\Delta t, 1/\Delta\tau+1/\Delta t]$. When $\Delta(Q^{n+1})^m=Q^{n+1,m+1}-Q^{n+1,m}$ is iterated to zero, the momentum equations are satisfied and the divergence of the velocity at time level $n+1$ is zero. It is noted that in Equation (24) no linearization is applied at each physical time step, and accurate solution in time is achieved at each physical time step when a steady state solution is attained in pseudotime. In order to expedite the time marching process in pseudotime, the pseudotime is usually set to a very large value, e.g. 10^{12} [14]. In the present study it is set to infinity without causing a stability problem.

In the time discretization described above the cell area V is treated as constant. For free surface flow calculations this is equivalent to assuming that the computational mesh is stationary during the stage of pseudotime iteration.

During the implicit time integration some computational work relating to the inversion process can be saved by letting the inviscid Jacobian on the implicit side of the equation to be first-order-accurate, and to simulate the viscous Jacobian using a Laplacian-like dissipation operator, i.e.

$$-\frac{1}{Re}\sum_{j=k(i)}F^{\mathrm{v}}_{i,j}\Delta l_j=-\frac{1}{Re}\sum_{j=k(i)}\nabla Q^{\mathrm{v}}_{i,j}\cdot\mathbf{n}\Delta S_j\cong-\frac{1}{Re}\sum_{j=k(i)}\frac{\mathbf{r}_{j,i}\cdot\mathbf{n}\Delta l_j}{|\mathbf{r}_{j,i}|^2}(Q^{\mathrm{v}}_j-Q^{\mathrm{v}}_i), \tag{25}$$

where $\mathbf{r}_{j,i}$ is the distance vector from cell centre i to cell centre j. The system of Equations (24) can be written in matrix form as

$$(D+L+U)\Delta Q^s=\mathrm{RHS}, \tag{26}$$

where D is a block diagonal matrix, L is a block lower triangular matrix, and U is a block upper triangular matrix. Each of the elements in D, L and U is a 3×3 matrix. An approximate LU factorization (ALU) scheme as proposed by Pan and Lomax [21] can be adopted to inverse the implicit equation (26) in the form

$$(D+L)D^{-1}(D+U)\Delta Q^s=\mathrm{RHS}, \tag{27}$$

where the error of the factorization is equal to $LD^{-1}U\Delta Q^s$. Within each time step of the implicit integration the subiteration is terminated when the L_2 norm of the iteration process

$$L_2 = \left\{ \left[\sum_{i=1}^{N} (Q^{s+1} - Q^s)^2 \right] / N \right\}^{1/2}, \tag{28}$$

is less than a specified limit.

6. NUMERICAL RESULTS

6.1. Oscillating flat-plate flow

As an initial test case, the unsteady flow over an oscillating infinite flat-plate was calculated. This problem, known as Stoke's second problem [22], has also been used by Pan *et al.* [20] and Rogers and Kwak [13] to verify their schemes which were based on body conforming grids.

In this problem, the plate is set to oscillate in the x-direction with a velocity

$$u_{\text{plate}} = u_o \cos(\omega t). \tag{29}$$

The analytical solution for the velocity profile over the plate is given by

$$u(y, t) = u_o \exp\left(-y \sqrt{\frac{\omega}{2\nu}} \right) \cos\left(\omega t - y \sqrt{\frac{\omega}{2\nu}} \right). \tag{30}$$

The computational mesh was extended to a height where the exact solution for the velocity is $< 5 \times 10^{-4} u_o$. The variation of mesh size is such that, at the plate surface, the cell size is equal to 0.02 of the total mesh height, while at the opposite end it is equal to the width of the domain. The velocity u_o was set to unity, the frequency was set to 2π, the viscosity was set to

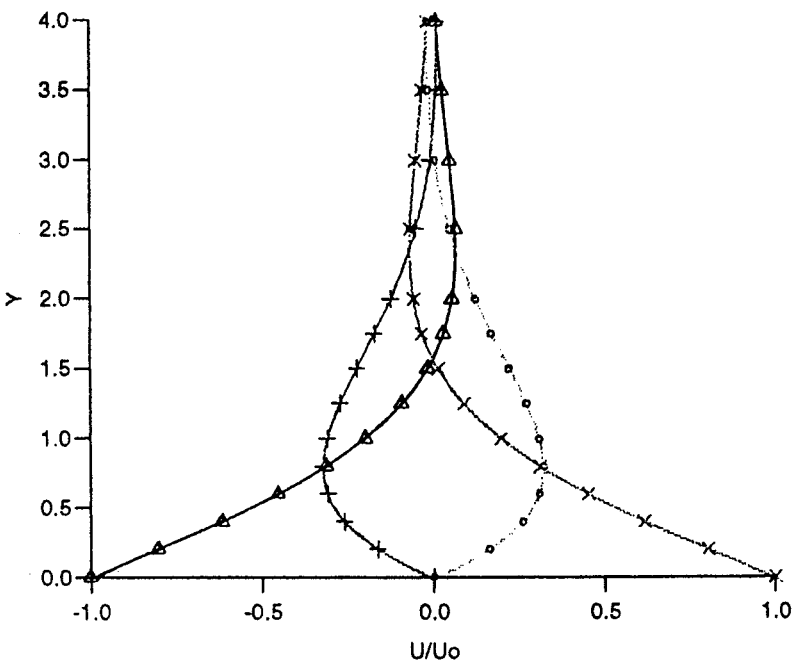

Figure 3. Velocity profiles for flow over an oscillating plate. Solid lines denote computed solution, symbols denote analytical solution. ○, $t = 8.25$; △, $t = 8.5$; +, $t = 8.75$; ×, $t = 9$.

π, the physical time step was set to 0.01 and the coefficient of pseudocompressibility, β was set to 100. At each time step, the root mean square (rms) of divergence of velocity and the residual at any point were set to be iterated to $< 1 \times 10^{-4}$. The problem was run until the velocity became fully periodic in time. The computed solution is compared with the analytic solution in Figure 3, where the solid lines denote the computed results and the symbols denote the analytical solution. Excellent agreement is obtained.

6.2. Inviscid flow in an one-dimensional channel

The problem of inviscid flow in a 1D channel with an oscillating back pressure was tackled using the present 2D code. This problem has been also used by Merkle and Athavale [11] and Rogers and Kwak [13] in order to verify their methods. The computational domain was triangulated with a layout of 25 cells in the x-direction and three cells in the y-direction. At the inflow of the channel, a constant total pressure head was specified, and the static pressure at the outflow was set to be

$$p_{\text{exit}} = p_o + p_e \sin(\omega t). \tag{31}$$

An analytical solution for this problem exists when the back pressure oscillation p_e is small compared with the mean back pressure p_o. In this example, the ratio of p_e to p_o was set to be 0.1 and the mean velocity was set to unity. The analytical solution following for this case is given by

$$u(t) = 1 - \frac{p_e}{1+\omega^2}[\sin(\omega t) - \omega \cos(\omega t) - \omega e^{-1}], \tag{32}$$

and

$$p(x, t) = p_o + p_e \sin(\omega t) + (x-1)\frac{p_e \omega}{1+\omega^2}[\cos(\omega t) + \omega \sin(\omega t) + e^{-1}]. \tag{33}$$

It can be noted from the above equations that both the pressure and the velocity solutions contain exponential decaying terms which correspond to the initial transient, and sine and cosine terms which correspond to the periodic harmonic motion. The analytical solution shows that the velocity is a function of time only, which indicates that the flow is incompressible and the channel area is constant.

The initial conditions for this problem were set using Equations (32) and (33) evaluated at $t = 0$. The frequency ω was set to 10, the physical time step was set equal to $\pi/(15\omega)$, and the pseudocompressibility coefficient, β was set to 100. At each time step, the rms of divergence of the velocity and the residual at any point were set to be iterated to below 1×10^{-6}. Figure 4 shows the computed velocity time series as compared with the analytical solution. Very good agreement is obtained.

6.3. Viscous vortex shedding behind a circular cylinder

Viscous vortex shedding behind a circular cylinder was simulated with the purpose to validate the code for a fully 2D flow. The problem was run using a mesh with its outer boundaries ten diameters from the cylinder. In this mesh, the cell size next to the cylinder surface is 0.045 diameters. The Reynolds number was set to 200, the physical time step was set to be 0.025 and the pseudocompressibility coefficient β was set to 10 for faster convergence. At each time step, the rms of divergence of velocity and the residual at any point were set to be iterated to $< 1 \times 10^{-4}$. The flow was started from stationary conditions and

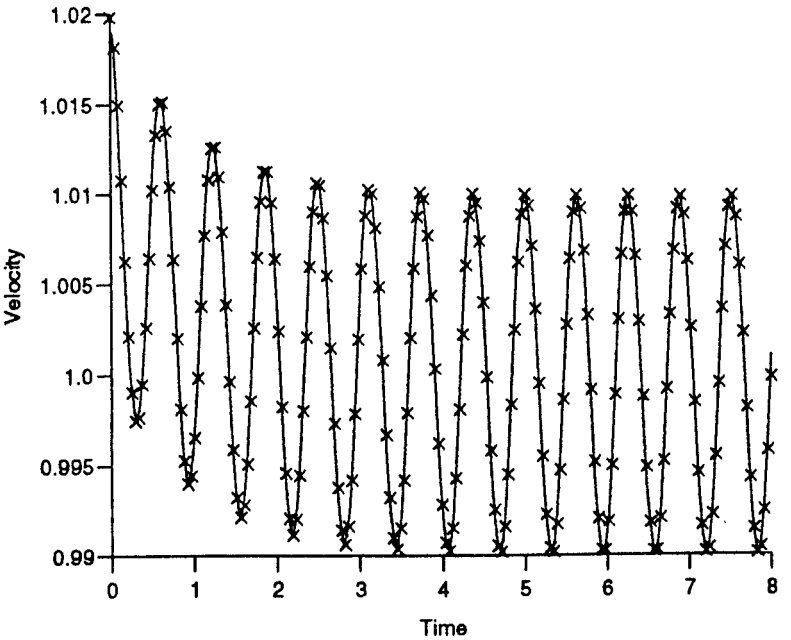

Figure 4. Time series of velocity for flow in an inviscid channel with an oscillating black pressure. Solid line denotes computed solution and + denotes analytical solution.

the simulation run until periodic shedding of vortices occurred. A pronounced asymmetric wake began to appear at the non-dimensional time of 40 and became completely periodic at a time of 90. Figure 5 shows the lift and drag coefficients on the cylinder versus time. The table below shows a comparison between the present results and those obtained by other researchers.

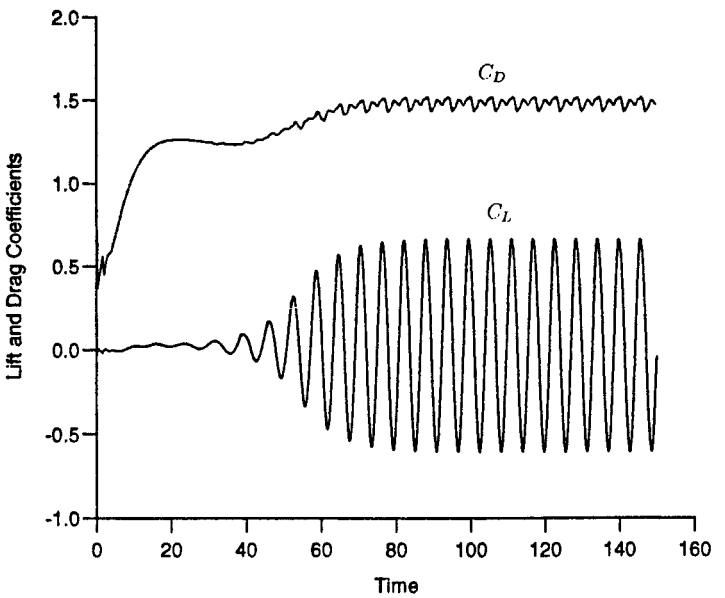

Figure 5. Lift and drag coefficients versus time for flow over a circular cylinder ($Re = 200$).

	C_D	C_L	St
Present	1.48 ± 0.05	± 0.63	0.183
Rogers *et al.* [13] 3rd order	1.29 ± 0.05	± 0.75	0.160
Rogers *et al.* [13] 5th order	1.23 ± 0.05	± 0.65	0.185
Lecointe *et al.* [23] 2nd order	1.46 ± 0.05	± 0.70	0.227
Lecointe *et al.* [23] 4th order	1.58 ± 0.0035	± 0.50	0.194

In the table above C_D is the drag coefficient, C_L is the lift coefficient, and St is the Strouhal number. In comparison with the experimental C_D value of 1.3 [13], the present numerical simulation is seen to perform reasonably well, but not as well as that of Rogers *et al.* [13]. The reason for this behaviour may be attributed to the fact that the computational mesh used next to the cylinder is more than one order coarser than that used by Rogers *et al.* [13]. Figure 6 shows the streamlines in sequence for one complete cycle of the Kàrmàn vortex shedding.

It is worth mentioning that the vortex shedding test was also run with $Re = 80$ and 60 using the same computational mesh and time step. Shedding for $Re = 80$ was fully captured with $C_D = 1.25$ and $C_L = 0.17$, while at $Re = 60$, the C_D being 1.25 and the maximum C_L approximately 0.002, only small wiggles were captured and the shedding amplitude decayed in time. These results indicate that the lowest Reynolds number at which vortex shedding starts is about 60 and that no vortex shedding can be captured by the present scheme for $Re < 60$.

6.4. Flow over a backward-facing step

Further validation of the model in calculating internal flow problems was carried out by computing the flow over a backward-facing step. The definition of the problem is given in Figure 7 where the reattachment length of the separation bubble was calculated for a range of laminar flows with $Re = 100$ to 800. The step height in this problem is equal to the inlet height, and the Reynolds number is based on two step heights. The upstream boundary is located at a distance of 11 steps from the step and the downstream boundary is located at a distance of 20 steps from the step.

The calculated reattachment length is shown in Figure 8, where it is compared with several other sets of results including the experimental data of Armaly *et al.* [24]. The presented results show that the present code performed very well for the entire range of values of the Reynolds number. It is observed that separation at the opposite wall occurs for $Re = 500$ and greater. This separation begins upstream of the reattachment point of the primary separation, and ends downstream of it. The size of this separation bubble ranges from four to six steps in height as the Reynolds number increases. This behaviour agreed reasonably well with the experimental observations which reported a range of five to eight step lengths. Figure 9 shows the velocity field at $Re = 500$.

6.5. Water column subjected to centrifugal force

In order to validate the code in simulating problems with a free surface, a simple hypothetical problem of a water column subjected to a centrifugal force was initially considered. This problem is similar to the case of a rotating fluid column studied by Thè *et al.* [25]

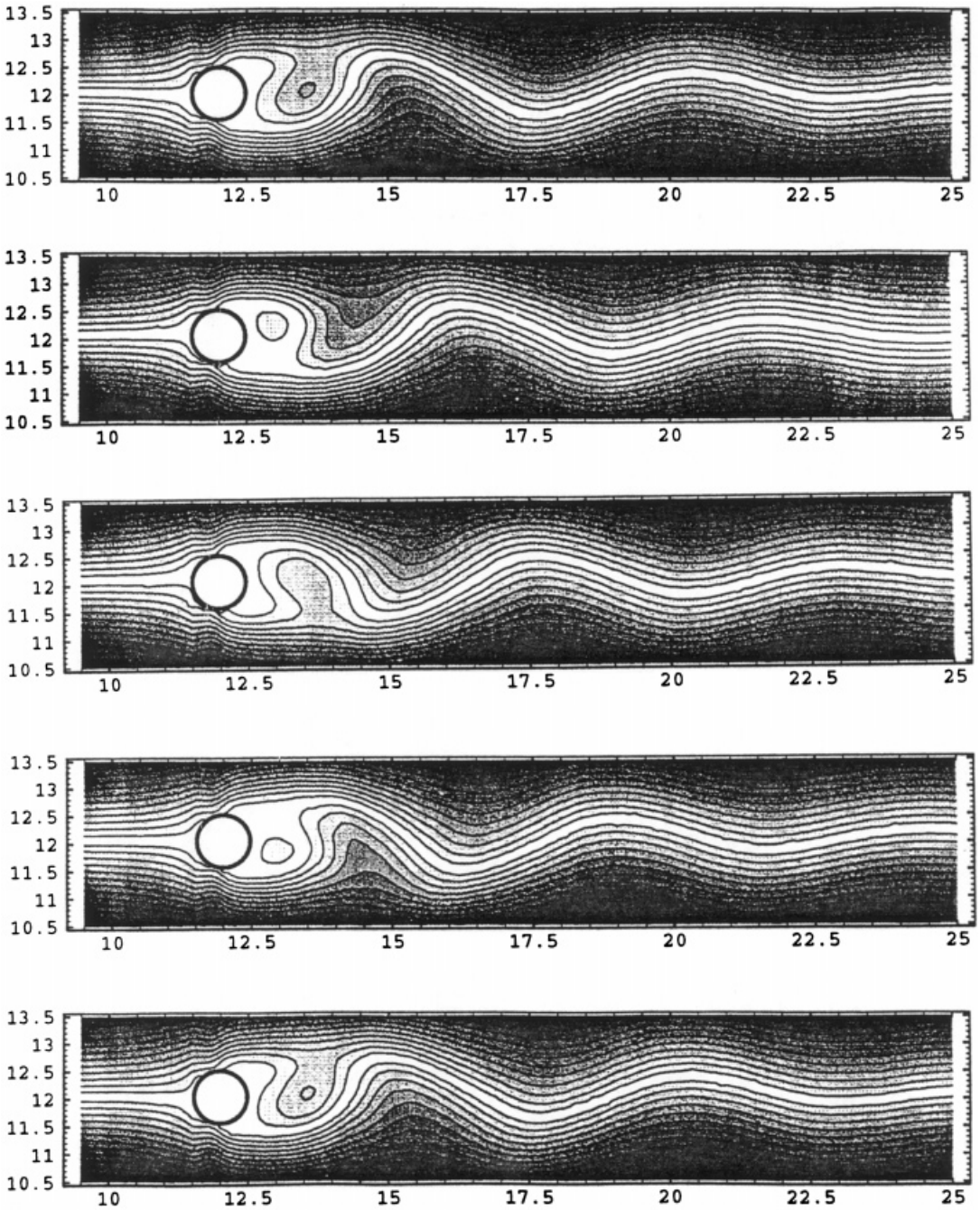

Figure 6. Streamline patterns showing in sequence (top to bottom) one cycle of the Kármán vortex shedding ($Re = 200$).

in verifying their surface-adaptive free surface code. The definition of the problem is given in Figure 10, where a centrifugal force applied laterally to the water column in an infinitely long channel is considered. The analytical solution of this problem is

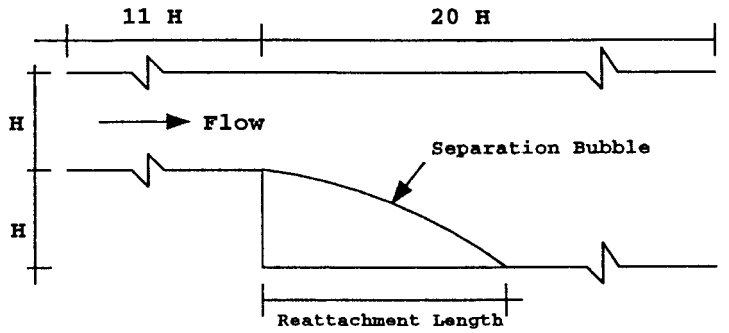

Figure 7. Problem definition for flow over a backward-facing step.

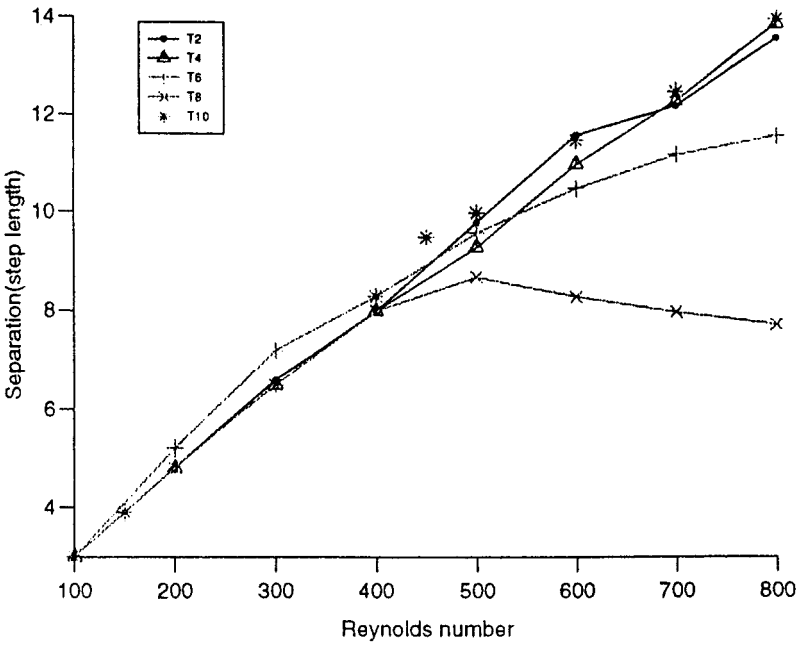

Figure 8. Reattachment length versus Reynolds number. T2, present results; T4, Rogers *et al.* [10]; T6, TEACH [24]; T8, Kim and Moin [10]; T10, Armaly *et al.* [24].

$$h_s(x) = h_{so} - \frac{(\omega L)^2}{2g}\left[\frac{1}{3} - \left(\frac{x}{L}\right)^2\right], \tag{34}$$

where h_s is the free water surface elevation, h_{so} is the initial water level, ω is a hypothetical angular frequency that generates the centrifugal force, L is the width of the water column and g is the gravitational acceleration.

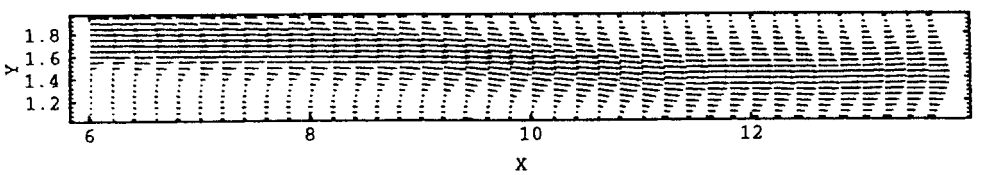

Figure 9. Velocity field for flow over a backward-facing step ($Re = 500$).

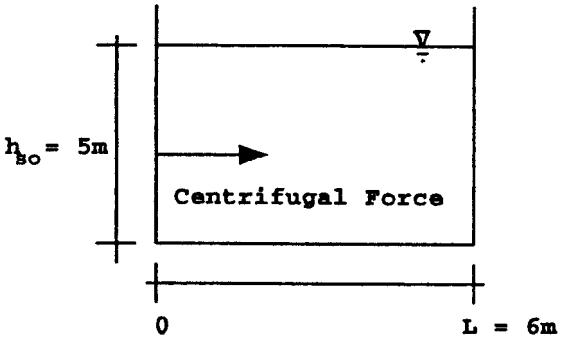

Figure 10. Problem definition for water column subjected to a centrifugal force.

In order to tackle this problem, h_{so} was set to 5 m, L was set to 6 m, ω was set to 1.476 rad s^{-1}, g was taken to be 9.81 m s^{-2}, and a non-slip wall condition was used. The problem domain was triangulated using mesh sizes ranging from 0.1 m at the free surface to 0.35 m at the bottom. The problem was run from the quiescent condition to steady state using a range of values of the leakage coefficients ε. It was found that when the material free surface condition is enforced, i.e. $\varepsilon = 0$, the movement of the free surface was damped heavily and the steady state solution was reached with very little oscillation. On the other hand, when $\varepsilon = 1$ was used, the free surface movement became pronounced and the steady state solution was reached after many cycles of oscillation. In all the runs, the free surface movements became negligible when the rms of velocity and pressure were iterated below 0.1. Figure 11 shows a comparison between the computed solution and the analytical solution.

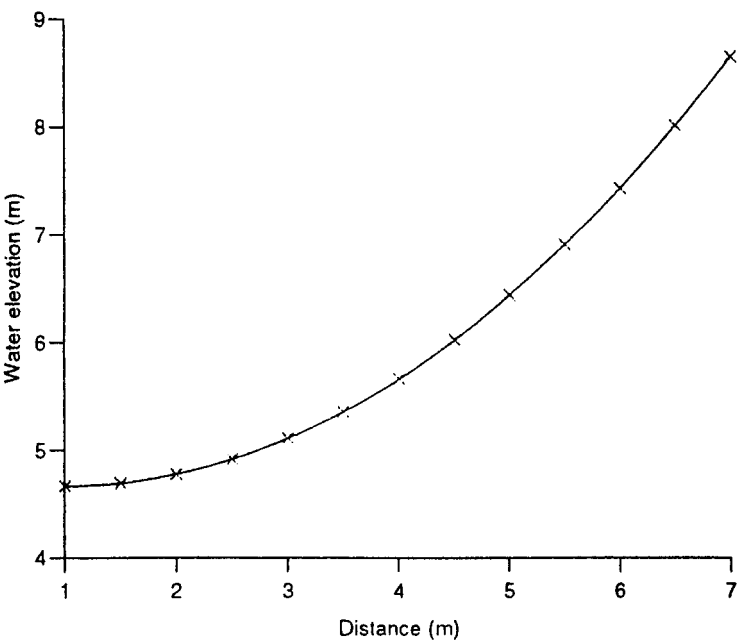

Figure 11. Water surface profile at steady state. Solid line, computed solution; $\times$, analytical solution.

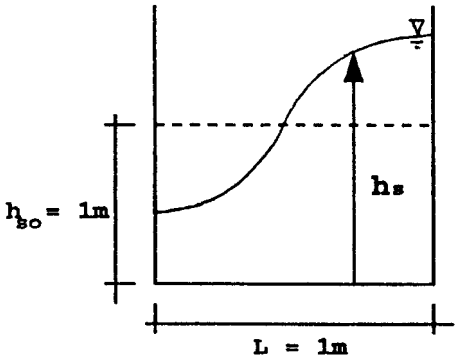

Figure 12. Problem definition for water oscillating in a tank: initial elevation.

6.6. Water oscillation in a tank

The capability of the code with regard to accurate time-dependent free surface calculations was further tested by simulating the evolution of the sinusoidal oscillating free surface shown in Figure 12. This case has also been studied by Thè *et al.* [25].

In this study, the mean water level h_{so} and the half wave length L were set to 1 m and the amplitude of the wave was set to 0.01 m. The flow was considered to be inviscid so that the oscillation in the tank should occur without significant decay. A slip-wall boundary condition was used. The computational domain was triangulated with a mesh size of 0.03 m distributed on the free surface. Several runs were conducted with different ε values. The time series of water elevation at the left and right walls of the tank were recorded, and are shown in Figure 13(a,b). It is noted from the figures that when ε is set to 0, the wave is critically damped, and its amplitude decays to zero without oscillation. However, it is also noted that when ε was set to 1, the oscillation became unstable and the wave amplitude diverged in time. Figure 13 (b) shows results corresponding to $\varepsilon = 0.993$. These results show that the oscillation was simulated stably with only a very small decay rate. The problem was rerun using $\varepsilon = 1$ but with a distribution of finer mesh at the free surface. It was found that, again, the oscillation was captured stably and the results obtained were almost identical to those of Figure 13 (b). This behaviour indicates that, for this particular case, a 'remedy' exists between the required fineness of mesh size at the free surface and the value of ε used.

6.7. Propagation of a solitary wave

In order to check whether the code can correctly calculate the wave speed and damping characteristics of a solitary wave traveling in a channel, the case studied by Tang *et al.* [26] and Thè *et al.* [25] was next considered. In this simulation the wave was set to travel in a channel of 20 m length, and h_o was equal to 1 m. The computational domain was triangulated with a mesh size equal to 0.1 m on the free surface. The wave celerity $C = \sqrt{gh_o}$ was set equal to 1 by taking $g = 1.0$ m s^{-2}. For this problem the Reynolds number is defined as $Re = h_o C/\nu$.

Initially, the simulation was carried out by setting the surface elevation, h, equal to the Boussinesq profile for a reflected wave with its peak amplitude, a_o at the left boundary of the domain, as

$$h = h_o + 2a_o \operatorname{sech}^2\left(\frac{x}{2}\sqrt{6a_o}\right). \tag{35}$$

The velocity components and the viscosity were set to zero. The problem was then run for 5 s and the resulting surface elevation and velocity field were stored as initial conditions for subsequent calculations. Figure 14 shows the development of the solitary wave during the initial 5 s of simulation.

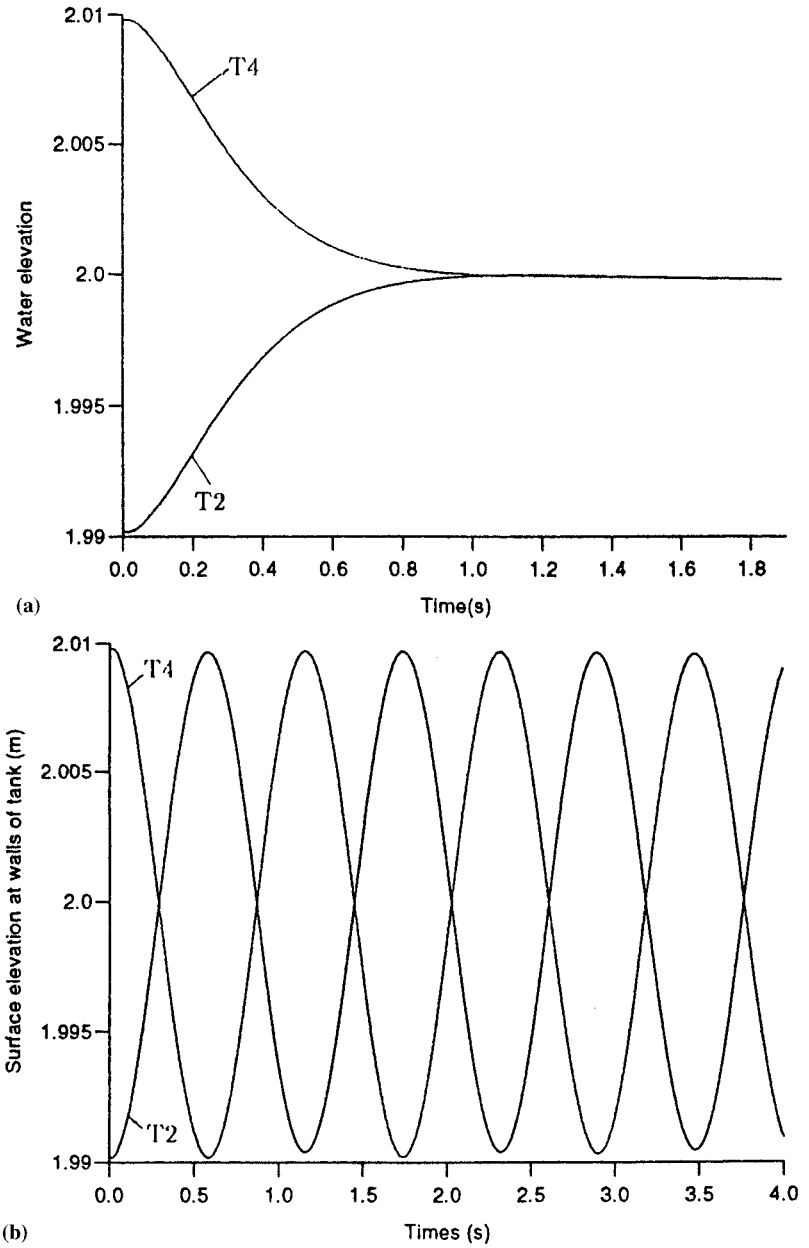

Figure 13. Surface elevation for case $\varepsilon = 0$. T2, left wall; T4, right wall. (b) Surface elevation for case $\varepsilon = 1$. T2, left wall; T4, right wall.

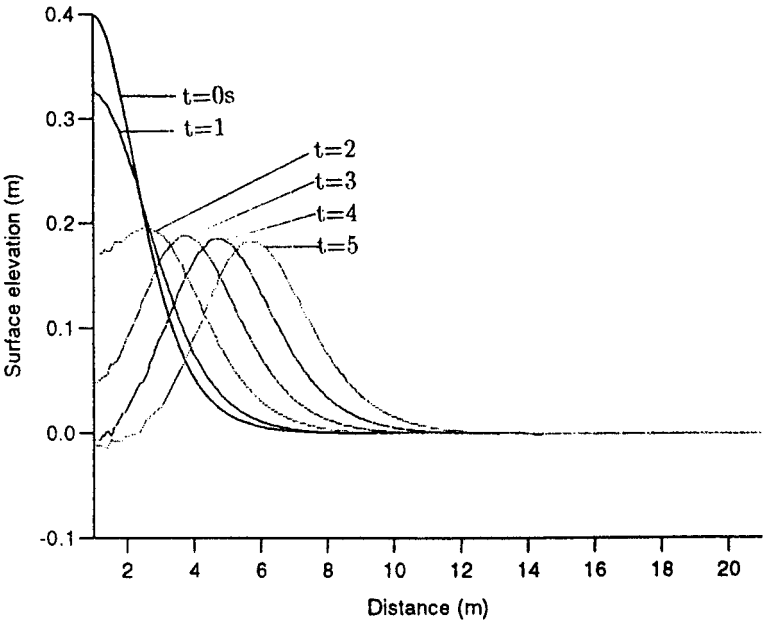

Figure 14. Development of the solitary wave.

The numerical experiments were carried on from 5 to 9 s with $Re = 5 \times 10^4$ and 50, respectively, with the coefficient ε set to 1. For $Re = 5 \times 10^4$ the amplitude decay rate was found to be -2×10^{-3} m m^{-1}, and the wave speed was very close to 1 m s^{-1}. These results agreed exactly with those obtained by Thè *et al.* The decay rate obtained by Tang *et al.* is -1.7×10^{-3} m m^{-1} and the value given by Mei [27] is -1.75×10^{-3} m m^{-1} using his linear perturbation theory. For $Re = 50$, the amplitude decay rate from the present scheme is -5×10^{-3} m m^{-1}, which is lower than the values of -6.9×10^{-3}, -7.2×10^{-3} and -6.2×10^{-3} m m^{-1}, given by Thè *et al.*, Tang *et al.* and Mei, respectively. Considering the coarseness of the mesh at the free surface, the lower value of the decay rate at low Reynolds numbers is quite reasonable. It is noted that this tendency can be dealt with by using a lower value of ε, which would have the effect of increasing the decay rate. Figure 15 shows snap-shots of propagation of the solitary wave at $Re = 5 \times 10^4$.

It is also noted from the presented results that wiggles are observed trailing the solitary wave. These wiggles are numerically spurious, they do not form part of the physical solution, and they can be quite easily suppressed using a finer computational mesh.

6.8. Flow over a submerged hydrofoil

In order to validate the code for more complicated free surface flows, i.e. for problems with a free surface that crosses the still water level more than once, the last problem dealt with in this work is the flow over a submerged hydrofoil. This problem has been studied in the laboratory by Duncan [28] and numerically by Hino *et al.* [16] who used their multigrid explicit cell-vertex scheme. As this problem requires rather long simulation times when tackled using the present single mesh code, only one case is simulated and presented here.

The hydrofoil section considered is the NACA0012 with an angle of attack of 5°. The depth of submergence measured at mid-chord and non-dimensionalized with respect to the chord

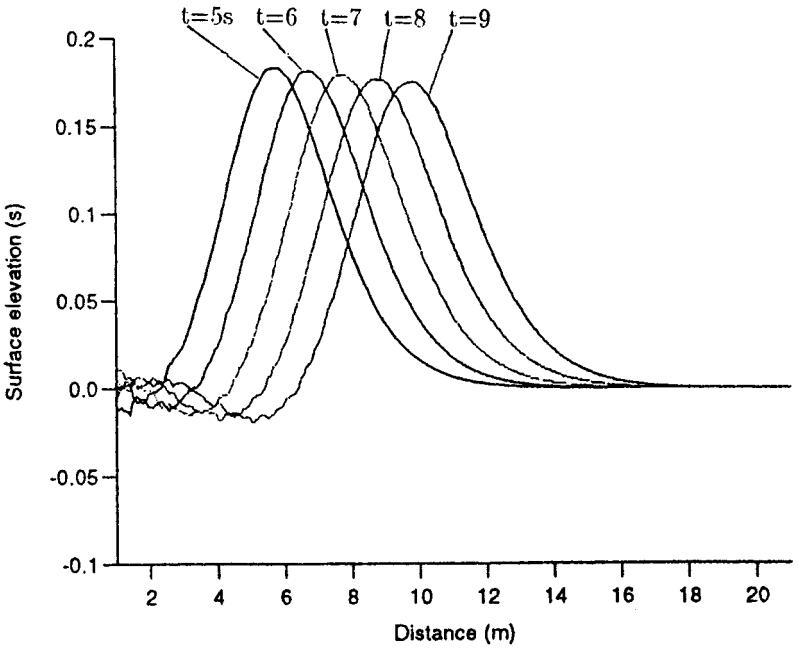

Figure 15. Propagation of the solitary wave ($Re = 5 \times 10^4$).

length is 1.034. The Froude number used is $F = 0.5672$ and the computational mesh is shown in Figure 16. The mesh consists of 7857 cells, and it is noted that this mesh is coarser than those used by Hino *et al.* [16], where the finest level of their three level meshes consists of 12854 cells. In order to prevent any reflection of waves into the solution domain, and to keep the outflow water elevation at still water level, an exponential damping term of the form

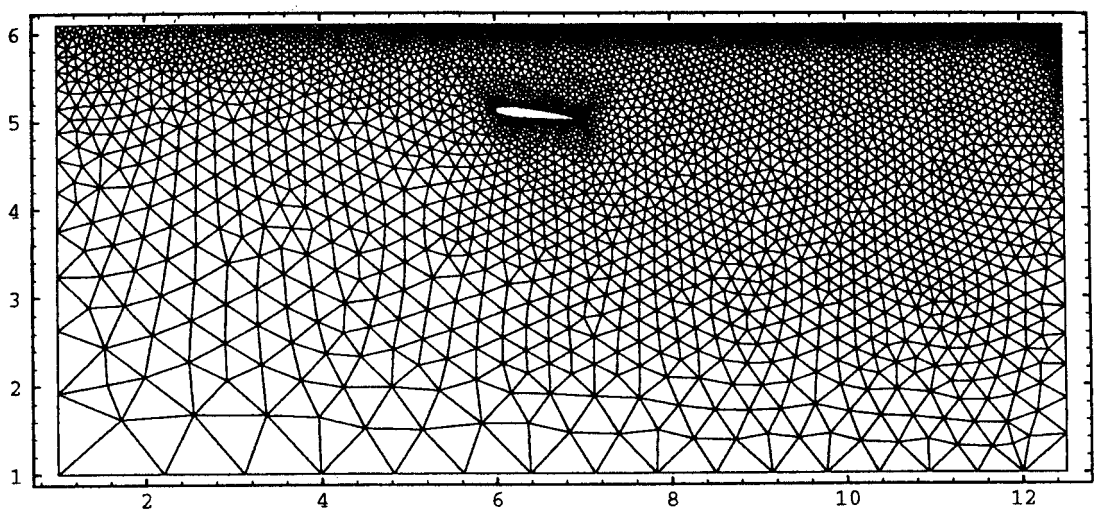

Figure 16. Computational mesh for flow over a submerged hydrofoil.

$$\gamma(x) = \begin{cases} 0.5 \exp\{\ln[s(x)+1](x-x_d)/(x_o-x_d)\} - 1 & \text{if } x_d \le x < x_o \\ 0 & \text{otherwise} \end{cases},$$

is added to the RHS of the free surface kinematic condition. In the above expression $s(x) = h(x)$, $h(x)$ is the water elevation with respect to the still water level, x_o is the x-co-ordinate of the outflow boundary, and x_d is defined as

$$x_d = x_o - \pi F^2/2. \tag{37}$$

Equation (37) equivalent to setting a damping zone that extends to one quarter of the deep-water wavelength, calculated from the linear theory, from the outflow boundary. The simulation run until steady state was attained, with the coefficient ε set to 1.

It is important to highlight that this problem was run twice, in the first instance using the formulation of Equation (2) with the vector $H = (0, -s_x/F_2, 0)^{-1}$, and also adopting the formulation of Hino *et al.* [16] whereby the vector H is nullified by being incorporated into the pressure term of the momentum equations and the pressure at the free surface is given by $p_o + h(x)/F^2$, where p_o denotes the atmospheric pressure. When the first formulation was used, it proved to be very difficult to obtain a stable free surface at the outflow boundary. A stable free surface was achieved only at the expense of applying very heavy damping at the outflow boundary, by setting $s(x) = v(x) + h(x)$ in Equation (36) and increasing the multiplier to one. The results so obtained were found to be quite inferior to those obtained by adopting the formulation of Hino *et al.* [16]. It is believed that this behaviour is due to the fact that in the first formulation the slope of the free surface is used directly as the driving force in the momentum equations, and the solution algorithm cannot handle the complication brought about by the damping introduced at the outflow boundary which destroys the true free surface slopes.Figure 17 shows results from the present scheme compared with those obtained by Hino *et al.* [16] as well as the experimental data of Duncan [28]. The units in the figure have been

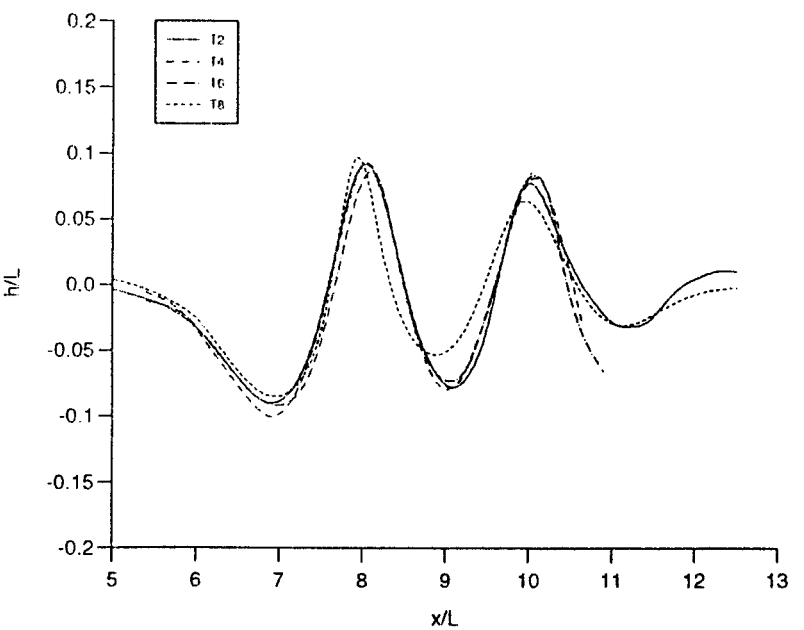

Figure 17. Wave profile over a submerged hydrofoil. T2, present scheme with the formulation of Hino *et al.* [16]; T8, present scheme with a formulation using the free surface slope; T4, Duncan [28]; T6, Hino *et al.* [16].

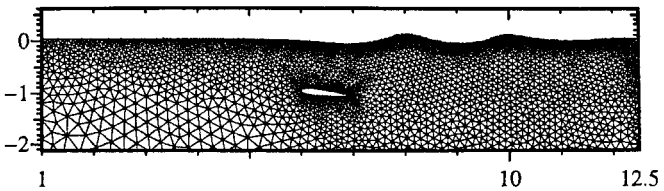

Figure 18. Computational mesh at steady state for flow over a submerged hydrofoil.

non-dimensionalized with respect to the hydrofoil length L. Figure 18 depicts the computational mesh near the free surface at steady state. The presented results show that the present scheme performs very satisfactorily.

7. CONCLUSIONS

A numerical solver of the incompressible Navier–Stokes equations for flows with or without a free surface has been developed and tested. The algorithm is based on a cell-centered finite volume method, it makes use of the method of pseudocompressibility, and is capable of dealing with both steady state and time-dependent problems. The method of pseudocompressibility allows the equations to be solved as a hyperbolic system in pseudotime. Accurate solutions at each time step are obtained by achieving a steady state solution in pseudotime. It has been found that the pseudotime step can be set to infinity without causing any stability problems. When applied to unstructured meshes consisting of triangular elements the algorithm is capable of efficiently handling flow fields with complex geometries and flows with a time-dependent free surface boundary. For flows with a free surface the computational mesh is adapted to the computed free surface at every time step. This is accomplished by applying a simple one-dimensional stretching technique to the mesh points, and during the process of pseudotime iteration the computational mesh is considered to be stationary. This approach simplifies the algorithm by allowing the time derivative of the computational cell area to be eliminated from the discretized equations. Any side effects arising from this approach are compensated by introducing a 'leakage coefficient' which relaxes the material free surface boundary condition by allowing the fluid particles to flow through the free surface during the course of numerical iteration. The numerical solver has been verified by application to a number of cases, ranging from very simple flows, where the numerical solutions agreed very well with available analytical solutions, to more complex flows, where the results were in close agreement with both experimental data and results from other reported numerical schemes. The results presented show that the present technique is robust.

REFERENCES

1. A. Jameson and D. Mavriplis, 'Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh', *AIAA Paper 85-0435*, 1985.
2. D. Mavriplis and A. Jameson, 'Multigrid solution of the Euler equations on unstructured and adaptive meshes', *AIAA Paper 87-0353*, 1987.
3. J.A. Desiseri and A. Dervieux, 'Compressible flow solvers using unstructured grids', *von Karman Institute Lecture Series 1988-05*, 7-11, 1988, pp. 1–115.
4. T.J. Barth and D.C. Jespersen, 'The design and application of upwind schemes on unstructured meshes', *AIAA Paper 89-0366*, 1989.
5. D. Pan and J. Cheng, 'A second-order upwind finite-volume method for the Euler solution on unstructured triangular meshes', *Int. J. Numer. Fluids*, **16**, 1079–1098 (1993).

6. K. Anastasiou and C.T. Chan, 'Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes', *Int. J. Numer. Fluids*, (1996), accepted for publication.
7. A.J. Chorin, 'A numerical method for solving incompressible viscous flow problems', *J. Comput. Phys.*, **2**, 12–26 (1967).
8. J.L.C. Chang and D. Kwak, 'On the method of pseudocompressibility for numerically solving incompressible flows', *AIAA Paper 84-0252*, 1984.
9. C.L. Merkle and P.Y.L. Tsai, 'Application of Runge–Kutta schemes to incompressible flow', *AIAA Paper 86-0553*, 1986.
10. S.E. Rogers, D. Kwak and U. Kaul, 'On the accuracy of the pseudocompressibility method in solving the incompressible Navier–Stokes equations', *Appl. Math. Model.*, **11**, 3–44 (1987).
11. C.L. Merkle and M. Athavale, 'Time-accurate unsteady incompressible flow algorithms based on artificial compressibility', *AIAA Paper 87-1137*, 1987.
12. W.Y. Soh and J.W. Doodrich, 'Unsteady solution of incompressible Navier–Stokes equations', *J. Comput. Phys.*, **79**, 113–134 (1988).
13. S.E. Rogers and D. Kwak, 'An upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations', *AIAA J.*, **28**, 253–262 (1990).
14. S.E. Rogers and D. Kwak, 'Steady and unsteady solutions of the incompressible Navier–Stokes equations', *AIAA J.*, **29**, 603–610 (1991).
15. D. Pan, C. Lu and J. Cheng, 'Incompressible flow solution on unstructured triangular meshes', *Numer. Heat Transf. Part B*, **26**, 207–224 (1994).
16. T. Hino, L. Martinelli and A. Jameson, 'A finite-volume method with unstructured grid for free surface flow simulations', *Proc. 26th Int. Conf. Num. Ship Hydro-dyn.*, Iowa City, IA, 1993, pp. 173–193.
17. P.L. Roe, 'Approximate Riemann solvers, parameter vectors, and difference schemes', *J. Comput. Phys.*, **43**, 357–372 (1981).
18. P.L. Roe, 'A basis for upwind differencing of the two-dimensional unsteady Euler equations', in K.W. Morton and M.J. Baines (eds), *Numerical Methods for Fluid Dynamics II*, 1985, pp. 55–80.
19. P.L. Roe, 'Characteristic based schemes for the Euler equations', *AA. Rev. Fluid Mech.*, **18**, 337–365 (1986).
20. D. Pan and S.R. Chakravarthy, 'Unified formulation for incompressible flows', *AIAA Paper 89-0122*, 1989.
21. D. Pan and H. Lomax, 'A new approximate LU factorization scheme for the Navier–Stokes equations', *AIAA J.*, **26**, 163–171 (1988).
22. F.M. White, *Viscous Fluid Flow*, McGraw-Hill, New York, 1974, pp. 148–149.
23. Y. Lecointe and J. Piquet, 'On the use of several compact methods for the study of unsteady incompressible viscous flow round a circular cylinder', *Comput. Fluids*, **12**, 255–280 (1984).
24. B.F. Armaly, F. Durst, J.C.F. Pereiraand and B. Schönung, 'Experimental and theoretical investigation of backward-facing step flow', *J. Fluid Mech.*, **127**, 473–496 (1983).
25. J.L. Thè, G.D. Raithby and G.D. Stubley, 'Surface-adaptive finite-volume method for solving free surface flows', *Numer. Heat Transf. Part B*, **26**, 367–380 (1994).
26. C.J. Tang, V.C. Patel and L. Landweber, 'Viscous effects on propagation and reflection of solitary waves in shallow channels', *J. Comput. Phys.*, **88**, 88–113 (1990).
27. C.C. Mei, *The Applied Dynamics of Ocean Surface Waves*, Wiley, New York, 1983.
28. J.H. Duncan, 'The breaking and non-breaking wave resistance of a two-dimensional hydrofoil', *J. Fluid Mech.*, **126**, 507–520 (1983).

