INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 23, 545–566 (1996)

HIGHER-ORDER SCHEMES FOR FREE SURFACE FLOWS WITH ARBITRARY CONFIGURATIONS

CARLOS M. LEMOS

Department of Physical Oceanography, Instituto Hidrografico, R. das Trinas 49, 1294 Lisbon Codex, Portugal

SUMMARY

The purpose of the present work was to evaluate the importance of formal accuracy and of the conservation property in the numerical computation of incompressible flows with arbitrary free boundaries, such as occur in wave-breaking problems. Four spatial discretization methods were implemented in a computer code based on the VOF method for tracking free surfaces: a non-conservative four-point scheme, the conservative quadratic upstream interpolation method, the conservative linear extrapolation method and a lower-order conservative scheme based on the power-law discretization. The performance of the four schemes was evaluated in three test problems: the propagation of a solitary wave of high amplitude, the propagation of an undular hydraulic jump and the flow resulting from a breaking hydraulic jump. The main conclusion obtained in the present work was that discrete momentum conservation is more important than the formal accuracy of the spatial discretization scheme, particularly when there is recirculation and breaking.

KEY WORDS: free surfaces; VOF method; higher-order schemes; conservation property

INTRODUCTION

The numerical computation of flow problems involving free surfaces of arbitrary configuration plays an important role in the analysis of many problems in coastal and oceanographic engineering, hydraulic engineering, aerospace engineering and mould-filling technology. Another closely related problem is the calculation of the hydrodynamic forces that arise from the interaction between jets or water waves with fixed or moving obstacles. However, the simulation of flows with multiple free boundaries remains a challenging topic in computational fluid dynamics because of the intrinsic difficulties of the problem: description of the evolution and shape of arbitrary free surfaces without logical hurdles, formulation of accurate numerical boundary conditions, recirculation due to the contortions of the free surface, turbulence and air–water mixing.

The marker-and-cell (MAC) method, introduced by Harlow and Welch,[1] was the first successful algorithm for the computation of incompressible flows involving arbitrary free surfaces. However, the MAC method had several important limitations: the treatment of the free surface was inefficient, the boundary conditions were approximate and the numerical scheme used was prone to instability. These limitations were partly overcome in later works. The volume-of-fluid (VOF) method, introduced by Nichols and Hirt,[2] allowed a more efficient tracking of the free surface, whereas the introduction of the partial volume function method[3] permitted a more accurate representation of internal boundaries without the need for introducing curvilinear co-ordinates. Other improvements

CCC 0271–2091/96/060545–22

Received June 1995
Revised January 1996

included the formulation of more accurate free surface boundary conditions[4,5] and the incorporation of a k–ε turbulence model.[6] These subsequent developments of the MAC method were mainly based on low-order non-conservative schemes.

The importance of using higher-order schemes to obtain accurate numerical solutions to problems involving complicated vorticity dynamics was demonstrated by Davies and Moore[7] in the context of confined flows. Thakur and Shyy[8] made a thorough comparison of several schemes in steady and confined flow problems and concluded that the discretizations based on the quadratic upstream interpolation method or the second-order upwind method are clearly superior to the central differencing or lower-order upwind schemes. These authors used the SIMPLE algorithm,[9] which leads naturally to conservative discretizations, as the basic building block for the implementation of the spatial discretizations. These developments concerning the testing and comparison of higher-order schemes for confined flows have not been extended to free surface flows, especially those involving breaking and recirculation.

In this work, four spatial discretization schemes were implemented in a computer programme based on the SOLA-VOF code,[10] with the purpose of evaluating the importance of the order of accuracy and discrete momentum conservation on the quality of the numerical solutions for three test problems of increasing complexity. The four numerical schemes considered were (i) a non-conservative scheme based on four-point discretizations of the derivatives in the advection operator, (ii) the conservative quadratic upstream interpolation method, (iii) the conservative linear extrapolation method and (iv) the lower-order (conservative) power-law discretization. The three test problems considered herein were the propagation of a solitary wave of high amplitude, the propagation of an undular hydraulic jump and the propagation of a breaking hydraulic jump.

In what follows, the contents of this paper will be outlined. The next section contains a description of the theoretical formulation (governing equations, grid arrangement and basic computational cycle). The next two sections deal with the spatial discretization schemes for the momentum equation and the treatment of the continuity equation respectively. The numerical treatment of the boundary conditions is described in the fifth section. The last two sections describe the application studies and the main conclusions respectively.

The most relevant conclusion obtained in the present work is that discrete momentum conservation is more important for the stability and accuracy of the numerical solutions than is the formal accuracy of the scheme. This appears to be especially true for problems involving free surface breaking and strong breaker-induced recirculation. For the test problems considered, the lower-order power-law discretization produced numerical solutions with almost the same quality as those obtained using the other (higher-order) conservative schemes. This result indicates that to take full advantage of higher-order schemes in free surface flow simulations, it is necessary to improve other parts of the algorithm, such as the time-stepping scheme and the numerical boundary conditions.

THEORETICAL FORMULATION

The 2D motion of an incompressible fluid with uniform properties in a domain Ω with boundary $\partial\Omega$ can be stated as follows: determine $\mathbf{u} = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ and $p(\mathbf{x}, t)$ $\forall \mathbf{x} \in \bar{\Omega} = \Omega + \partial\Omega$ for $t > 0$ such that

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \nabla^2 \mathbf{u} = -\frac{1}{\rho} \nabla p + \mathbf{g}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where t is the time, $\mathbf{u}$ is the velocity, p is the pressure, ρ is the (constant) density, ν is the coefficient of kinematic viscosity, $\mathbf{g} = (g_x, g_y)$ is the vector of the external force field and ∇ is the nabla operator. An alternative form of the momentum equation is the so-called conservation form

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u} - \mu\nabla\mathbf{u}) = -\nabla p + \rho\mathbf{g}, \tag{1a}$$

where $\mu = \rho\nu$ is the coefficient of dynamic viscosity. Equation (1a) is appropriate for the derivation of numerical schemes that ensure conservation of momentum in a discrete sense.[11]

The existence of a free surface implies that the position of the boundary $\partial\Omega$ is not in general known *a priori* and has to be determined as part of the solution. The fluid configuration can be described in terms of a volume-of-fluid (VOF) function $F(\mathbf{x}, t)$, whose value is unity at any point occupied by the fluid and zero elsewhere.[10] Cells with $F = 1$ are treated as full (i.e. interior) cells, whereas cells with $F = 0$ are considered empty. Cells with F-values between zero and unity which have at least one empty neighbour are treated as surface cells. The equation governing the time evolution of F is

$$\frac{\partial F}{\partial t} + \nabla \cdot (\mathbf{u}F) = 0, \tag{3}$$

which states that F moves with the fluid.

To define a well-posed initial–boundary value problem, the governing equations must be supplemented by appropriate initial and boundary conditions. The boundary conditions for $\mathbf{u}$ may be of either Dirichlet or Neumann type, i.e.

$$\mathbf{u} = \mathbf{u}_b(\mathbf{x}, t) \quad \forall\mathbf{x} \in \Gamma_D \tag{4}$$

or

$$-p + \mu\frac{\partial u_n}{\partial n} = f_n(\mathbf{x}, t) \quad \text{and} \quad \mu\frac{\partial u_\tau}{\partial n} = f_\tau(\mathbf{x}, t) \quad \forall\mathbf{x} \in \Gamma_N, \tag{5}$$

where $\partial\Omega = \Gamma_D + \Gamma_N$, Γ_D and Γ_N are the parts of the boundary where Dirichlet and Neumann boundary conditions are imposed respectively and n and τ denote the normal and tangential directions respectively. These conditions correspond to either velocity or stress specified on the boundary respectively. The boundary conditions for F must be specified on every boundary point for which $\mathbf{u}_b \cdot \mathbf{n} < 0$.

The initial condition for $\mathbf{u}$ must satisfy the continuity equation on Ω and its normal component must be continuous on $\partial\Omega$. The initial F-function distribution must be specified in such a way that for the initial fluid configuration $F = 1$ in full cells and the fractional volume in the surface cells defines the correct interface shape.

In low-speed flows there is no evolution equation for p and the pressure field must be determined indirectly via the continuity equation. If $\mathbf{u}^*$ is a velocity field obtained using a consistent scheme $\mathbf{Q}$ for the time variation, advection and diffusion terms in equation (1) or (1a) with a provisional pressure field p^n,

$$\mathbf{u}^* = \mathbf{Q}(\Delta t, \Delta\mathbf{x}, \mathbf{u}^n) - \frac{\Delta t}{\rho}\nabla p^n,$$

then the pressure correction $\delta p = p^{n+1} - p^n$ satisfies a Poisson equation

$$\nabla\cdot\left(\frac{\Delta t}{\rho}\nabla[\delta p]\right) = \nabla \cdot \mathbf{u}^*. \tag{6}$$

This equation must be supplemented by Neumann conditions on Γ_D (obtained by projecting the momentum equation onto the direction of the normal $\mathbf{n}$) and Dirichlet conditions on Γ_N (e.g. on free surface boundaries).[12] Once the pressure correction is determined, the provisional pressure field is corrected and the provisional velocity field is updated to reflect the pressure change:

$$p^{n+1} = p^n + \delta p, \tag{7}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \frac{\Delta t}{\rho}\nabla[\delta p]. \tag{8}$$

The governing equations were discretized using a staggered mesh of rectangular cells of variable width Δx_i and height Δy_j. The arrangement of the dependent variables in a typical cell is illustrated in Figure 1. To derive conservative discretizations for equations (1a), (6) and (3), the computational domain is divided into a number of control volumes, on each of which the governing equations are integrated using the divergence theorem. The geometrical information required for the derivation of the discretized equation for a particular node P consists of the volume of the cell, the areas of the cell faces and the spacings between node P and its neighbours (see Figure 2). Owing to the staggered nature of the grid, the control volumes for u and v are shifted with respect to the control volumes for the scalar quantities.

The solution algorithm consists of three main steps. In the first step a provisional velocity field $\mathbf{u}^* = (u^*, v^*)$ is computed using a consistent scheme for the momentum equation with the old pressure gradient ∇p^n. This provisional velocity will not in general satisfy the continuity equation, because the advanced time pressure field is not yet known. In the second step the pressure correction $\delta p = p^{n+1} - p^n$ is calculated in such a way that the residual divergence left in the temporary velocity field is driven to zero. Once the pressure correction is available, the pressure p^n and the provisional velocity $\mathbf{u}^*$ are corrected so that $\mathbf{u}^{n+1}$ satisfies the continuity constraint. In the third step the fluid configuration is updated by solving equation (3), following the method of Nichols and Hirt which is described in detail in References 2 and 10.

MOMENTUM EQUATION APPROXIMATIONS

In the present work, three different higher-order spatial discretization schemes were considered: a non-conservative scheme based on four-point approximations for the advection term (scheme 1), a conservative scheme based on quadratic upstream interpolation (scheme 2) and a scheme based on linear extrapolation (scheme 3). For the purpose of evaluating the performance of the higher-order

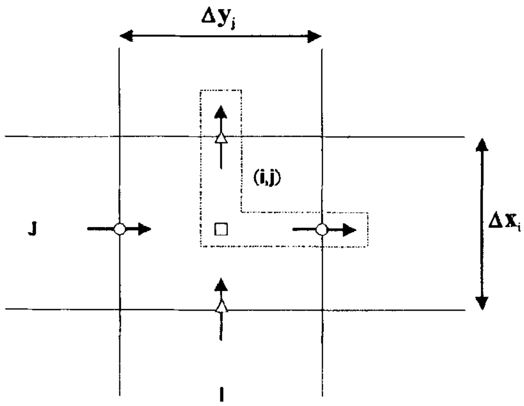


Figure 1. Location of dependent variables in typical mesh cell: ○, U-velocity; △, V-velocity; □, pressure and F-function

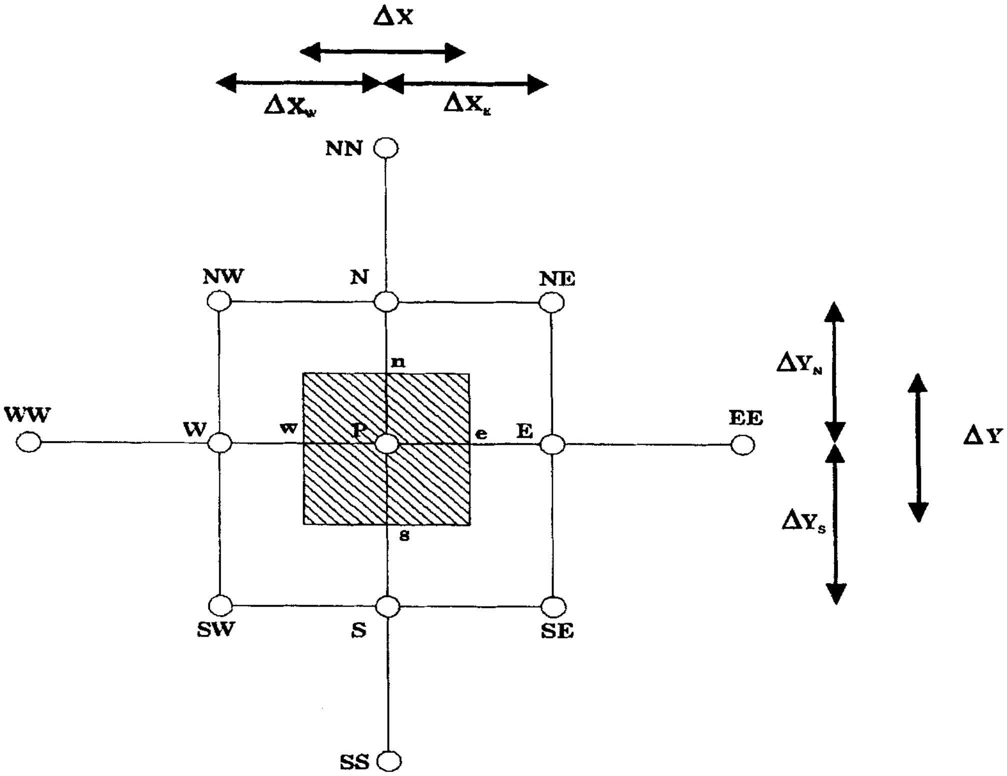


Figure 2. Schematization of control volume used for derivation of conservative discretizations

schemes with respect to their accuracy, stability and computational efficiency, a lower-order five-point conservative scheme based on the power-law discretization (scheme 4) was also implemented.

Scheme 1 is based on equation (1). For a generic node P the acceleration at time step n can be written as

$$\left(\frac{\partial\phi}{\partial t}\right)_{\mathrm{P}}^{n}=\left[-u\frac{\partial\phi}{\partial x}-v\frac{\partial\phi}{\partial y}+\nu\left(\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}\right)+S_\phi\right]_{\mathrm{P}}^{n},$$

with $\phi=u$ or v and $S_\phi=-(1/\rho)\partial p/\partial x+g_x$ or $-(1/\rho)\partial p/\partial y+g_y$ respectively. Following the usual procedure, the pressure gradient and viscous acceleration terms were approximated using centred differences. The partial derivatives $\partial\phi/\partial x$ and $\partial\phi/\partial y$ were evaluated using four-point approximations involving node P, two upstream nodes and one downstream node. The expression for $\partial\phi/\partial x$ is

$$\begin{aligned}\frac{\partial\phi}{\partial x}\approx{}&\frac{\Delta x_{\mathrm{W}}(\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{WW}})}{\Delta x_{\mathrm{E}}(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{W}})(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{WW}})}\phi_{\mathrm{E}}\\&+\frac{(\Delta x_{\mathrm{E}}-\Delta x_{\mathrm{W}})(\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{WW}})+\Delta x_{\mathrm{E}}\Delta x_{\mathrm{W}}}{\Delta x_{\mathrm{E}}\Delta x_{\mathrm{W}}(\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{WW}})}\phi_{\mathrm{P}}\\&-\frac{\Delta x_{\mathrm{E}}(\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{WW}})}{\Delta x_{\mathrm{W}}\Delta x_{\mathrm{WW}}(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{W}})}\phi_{\mathrm{W}}\\&+\frac{\Delta x_{\mathrm{E}}\Delta x_{\mathrm{W}}}{\Delta x_{\mathrm{WW}}(\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{WW}})(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{WW}})}\phi_{\mathrm{WW}}\end{aligned}\tag{9a}$$

if $u \geqslant 0$ and

$$\begin{aligned}\frac{\partial \phi}{\partial x} \approx & -\frac{\Delta x_{\mathrm{E}}(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{EE}})}{\Delta x_{\mathrm{W}}(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{W}})(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{EE}})}\phi_{\mathrm{W}} \\ & -\frac{(\Delta x_{\mathrm{W}}-\Delta x_{\mathrm{E}})(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{EE}})+\Delta x_{\mathrm{E}}\Delta x_{\mathrm{W}}}{\Delta x_{\mathrm{W}}\Delta x_{\mathrm{E}}(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{EE}})}\phi_{\mathrm{P}} \\ & +\frac{\Delta x_{\mathrm{W}}(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{EE}})}{\Delta x_{\mathrm{E}}\Delta x_{\mathrm{EE}}(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{W}})}\phi_{\mathrm{E}} \\ & -\frac{\Delta x_{\mathrm{E}}\Delta x_{\mathrm{W}}}{\Delta x_{\mathrm{EE}}(\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{EE}})(\Delta x_{\mathrm{W}}+\Delta x_{\mathrm{E}}+\Delta x_{\mathrm{EE}})}\phi_{\mathrm{EE}} \end{aligned} \tag{9b}$$

if $u < 0$, where the spacings are as shown in Figure 2. The approximations for $\partial\phi/\partial y$, which depend on the sign of v, are constructed in the same way. Equations (9a) and (9b) can be derived from Taylor series expansions and take into account the variation in the mesh spacing. If the mesh is uniform, then equations (9a) and (9b) reduce to

$$\partial\phi/\partial x \approx (2\phi_{\mathrm{E}}+3\phi_{\mathrm{P}}-6\phi_{\mathrm{W}}+\phi_{\mathrm{WW}})/6\Delta x$$

and

$$\partial\phi/\partial x \approx -(2\phi_{\mathrm{W}}+3\phi_{\mathrm{P}}-6\phi_{\mathrm{E}}+\phi_{\mathrm{EE}})/6\Delta x$$

respectively, which are formally third-order-accurate.

For nodes adjacent to the mesh, free surface and internal boundaries the values ϕ_{EE}, ϕ_{WW}, ϕ_{NN} and ϕ_{SS} may not be available. Therefore, when computing the approximations for $\partial\phi/\partial x$ or $\partial\phi/\partial y$ for node P, it is necessary to test whether the nodal values required for the higher-order approximation are outside the computational domain, inside a void region or inside an internal obstacle. If one of these conditions is verified, than a simple first-order upwind differencing is used instead of (9a) or (9b).

The conservative schemes 2–4 are based on equation (1a) and were constructed as follows. Combining the momentum and continuity equations, the following expression is obtained for the acceleration within the control volume enclosing a generic node P:

$$\frac{\partial\bar{\phi}_{\mathrm{P}}^{n}}{\partial t}(\rho_{\mathrm{P}}Vol) = (J_{\mathrm{w}}-J_{\mathrm{e}}+J_{\mathrm{s}}-J_{\mathrm{n}})^{n}+(\bar{S}_{\phi})Vol, \tag{10}$$

where $\phi = u$ or v, an overbar denotes the average over the control volume, *Vol* is the volume of the ϕ_{P}-cell and J_{w}, J_{e}, J_{n} and J_{s} are the momentum fluxes due to advection and diffusion through the faces of the control volume (see Figure 2). These fluxes may be expressed in terms of the neighbour nodal values as

$$J_{\mathrm{w}} = F_{\mathrm{w}}\phi_{\mathrm{w}} - D_{\mathrm{w}}(\phi_{\mathrm{P}}-\phi_{\mathrm{W}}), \tag{11a}$$

$$J_{\mathrm{e}} = F_{\mathrm{e}}\phi_{\mathrm{e}} - D_{\mathrm{e}}(\phi_{\mathrm{E}}-\phi_{\mathrm{P}}), \tag{11b}$$

$$J_{\mathrm{n}} = F_{\mathrm{n}}\phi_{\mathrm{n}} - D_{\mathrm{n}}(\phi_{\mathrm{N}}-\phi_{\mathrm{P}}), \tag{11c}$$

$$J_{\mathrm{s}} = F_{\mathrm{s}}\phi_{\mathrm{s}} - D_{\mathrm{s}}(\phi_{\mathrm{P}}-\phi_{\mathrm{S}}), \tag{11d}$$

where $F_{\mathrm{w}} = \rho_{\mathrm{w}}u_{\mathrm{w}}\Delta y_j$, $F_{\mathrm{e}} = \rho_{\mathrm{e}}u_{\mathrm{e}}\Delta y_j$, $F_{\mathrm{n}} = \rho_{\mathrm{n}}v_{\mathrm{n}}\Delta x_i$ and $F_{\mathrm{n}} = \rho_{\mathrm{n}}v_{\mathrm{n}}\Delta x_i$ are the mass flow rates across the control volume faces and $D_{\mathrm{w}} = \mu\Delta y_j/\Delta x_{\mathrm{W}}$, $D_{\mathrm{e}} = \mu\Delta y_j/\Delta x_{\mathrm{E}}$, $D_{\mathrm{s}} = \mu\Delta x_i/\Delta y_{\mathrm{s}}$ and $D_{\mathrm{n}} = \mu\Delta x_i/\Delta y_{\mathrm{N}}$ are the diffusion coefficients. The velocity and density at the cell faces are linearly interpolated when they are required at locations where they are not defined. Note that even though the density is uniform, the total mass (hence the average density) within a cell depends on the F-value in that cell. Schemes 2–4 differ only in the way the values of ϕ at the cell faces are obtained. Once the interpolation method for ϕ is specified, equation (10) can be cast into the form

$$\frac{\partial \bar{\phi}_\mathrm{P}^n}{\partial t} = \frac{1}{\rho_\mathrm{P} Vol}(a_\mathrm{E}\phi_\mathrm{E} + a_\mathrm{W}\phi_\mathrm{W} + a_\mathrm{N}\phi_\mathrm{N} + a_\mathrm{S}\phi_\mathrm{S} - a_\mathrm{P}\phi_\mathrm{P} + SU_\mathrm{P})^n, \tag{12}$$

where the coefficients a_E, a_W, a_N and a_S may be calculated in terms of the mass flow rates and diffusion coefficients, $a_\mathrm{P} = a_\mathrm{E} + a_\mathrm{W} + a_\mathrm{N} + a_\mathrm{S}$ and SU is the source term integrated over the control volume.

In *scheme 2* the cell face values were calculated using quadratic upstream interpolation.[13] For instance, if $u_\mathrm{e} \geqslant 0$, a local expansion of the form

$$\phi(x) = \phi_\mathrm{P} + k_1 x + k_2 x^2$$

is assumed between nodes W, P and E (see Figure 3, top). The undetermined coefficients are then

$$k_1 = \frac{\Delta x_\mathrm{E}}{\Delta x_\mathrm{T}\Delta x_\mathrm{W}}\Delta\phi_\mathrm{W} + \frac{\Delta x_\mathrm{W}}{\Delta x_\mathrm{T}\Delta x_\mathrm{E}}\Delta\phi_\mathrm{E}, \tag{13a}$$

$$k_2 = \frac{1}{\Delta x_\mathrm{T}\Delta x_\mathrm{E}}\Delta\phi_\mathrm{E} - \frac{1}{\Delta x_\mathrm{T}\Delta x_\mathrm{W}}\Delta\phi_\mathrm{W}; \tag{13b}$$

thus

$$\begin{aligned}\phi_\mathrm{e} &= \phi_\mathrm{P} + \frac{1}{4}\left(1 + \frac{\Delta x_\mathrm{W}}{\Delta x_\mathrm{T}}\right)\Delta\phi_\mathrm{E} + \frac{\Delta x_\mathrm{E}^2}{4\Delta x_\mathrm{T}\Delta x_\mathrm{W}}\Delta\phi_\mathrm{W} \\ &= \phi_\mathrm{up} + \frac{1}{4}\left(1 + \frac{\Delta x_\mathrm{W}}{\Delta x_\mathrm{T}}\right)\Delta\phi_\mathrm{E} + \frac{\Delta x_\mathrm{E}^2}{4\Delta x_\mathrm{T}\Delta x_\mathrm{W}}\Delta\phi_\mathrm{W},\end{aligned} \tag{14a}$$

where ϕ_up denotes the value of ϕ upstream of the east face, $\Delta\phi_\mathrm{E} = \phi_\mathrm{E} - \phi_\mathrm{P}$ and $\Delta\phi_\mathrm{W} = \phi_\mathrm{P} - \phi_\mathrm{W}$. If $u_\mathrm{e} < 0$, the local quadratic approximation is based on the nodes P, E and EE (Figure 3, bottom). In this case the cell face value was calculated using the expression

$$\phi_\mathrm{e} = \phi_\mathrm{up} - \frac{1}{4}\left(1 + \frac{\Delta x_\mathrm{EE}}{\Delta x_\mathrm{T}}\right)\Delta\phi_\mathrm{E} - \frac{\Delta x_\mathrm{E}^2}{4\Delta x_\mathrm{T}\Delta x_\mathrm{EE}}\Delta\phi_\mathrm{EE}, \tag{14b}$$

where $\phi_\mathrm{up} = \phi_\mathrm{E}$ and $\Delta\phi_\mathrm{EE} = \phi_\mathrm{EE} - \phi_\mathrm{E}$. The values of ϕ at the remaining cell faces were found using similar expressions.

During the development of the present work a more complete expansion of the form

$$\phi(x, y) = \phi_\mathrm{P} + k_1 x + k_2 x^2 + k_3 y + k_4 y^2 \tag{15}$$

was also tried. For the east face and for the case of $u_\mathrm{e} \geqslant 0$ the resulting ϕ_e-value is

$$\begin{aligned}\phi_\mathrm{e} = \phi_\mathrm{P} &+ \frac{\Delta x_\mathrm{E}^2}{4\Delta x_\mathrm{T}\Delta x_\mathrm{N}}(\phi_\mathrm{P} - \phi_\mathrm{W}) + \frac{1}{4}\left(1 + \frac{\Delta x_\mathrm{W}}{\Delta x_\mathrm{T}}\right)(\phi_\mathrm{E} - \phi_\mathrm{P}) + \frac{\delta_1}{4\Delta y_\mathrm{T}\Delta y_\mathrm{S}}(\phi_\mathrm{P} - \phi_\mathrm{S}) \\ &+ \frac{\delta_2}{4\Delta y_\mathrm{T}\Delta y_\mathrm{N}}(\phi_\mathrm{N} - \phi_\mathrm{P}),\end{aligned}$$

where $\delta_1 = \delta\Delta y_\mathrm{N} - 1/3(\delta^2 + \Delta y_\mathrm{N}\Delta y_\mathrm{S})$, $\delta_2 = \delta\Delta y_\mathrm{S} + 1/3(\delta^2 + \Delta y_\mathrm{N}\Delta y_\mathrm{S})$ and $\delta = \Delta y_\mathrm{N} - \Delta y_\mathrm{S}$. However, this scheme did not give better results than the simpler unidimensional quadratic interpolation method.

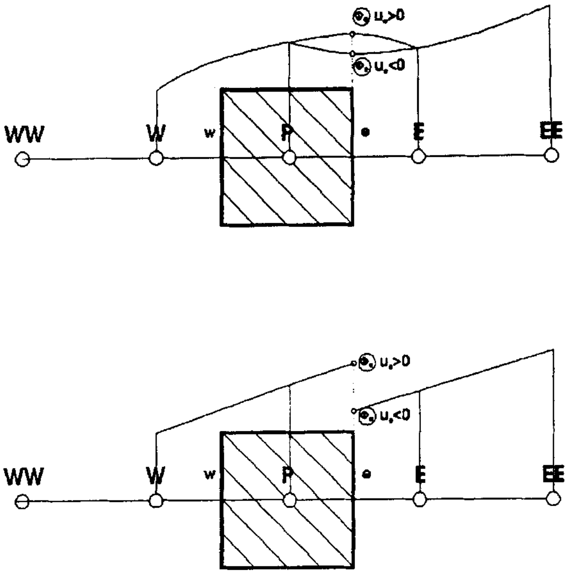


Figure 3. Schematic illustration of calculation of cell face value ϕ_e using quadratic upstream interpolation (top) and linear extrapolation (bottom)

In *scheme 3* the cell face velocities were obtained by means of linear extrapolation, which gives the second-order upwind method if the velocity field is uniform. If $u_e \geqslant 0$, the interpolated value of ϕ_e is

$$\phi_e = \phi_{up} + \frac{1}{2}(\phi_P - \phi_W)\frac{\Delta x_E}{\Delta x_W}, \tag{16a}$$

with $\phi_{up} = \phi_P$, whereas if $u_e < 0$, then

$$\phi_e = \phi_{up} - \frac{1}{2}(\phi_{EE} - \phi_E)\frac{\Delta x_E}{\Delta x_{EE}}, \tag{16b}$$

with $\phi_{up} = \phi_E$.

Schemes 2 and 3 were implemented as follows. For each ϕ-cell containing fluid, the influence coefficients were calculated using the expressions

$$a_E = \max(0, -F_e) + D_e, \tag{17a}$$

$$a_W = \max(0, F_w) + D_w, \tag{17b}$$

$$a_N = \max(0, -F_n) + D_n, \tag{17c}$$

$$a_S = \max(0, F_s) + D_s, \tag{17d}$$

$$a_P = a_E + a_W + a_N + a_S. \tag{17e}$$

This choice of the coefficients is equivalent to the use of $\phi = \phi_{up}$ at all cell faces[9] (i.e. to the first-order upwind scheme). The contributions due to the differences between the values of ϕ at the neighbour nodes were then added to the source term *SU*. This arrangement of the calculations has several advantages. First, the first-order upwind scheme is automatically selected if the addition of the correction terms to *SU* is skipped. This facilitates the selection of the interpolation method for cells adjacent to internal obstacles or free surface boundaries, as well as the treatment of the boundary conditions. Second, both the quadratic upstream interpolation and linear extrapolation methods can be implemented using the same basic routine. Only the code segments for adding the correction terms are different. Finally, this procedure allows the reformulation of the scheme as an implicit method if the coefficients a_E, a_W, a_N, a_S, a_P and the source term SU are stored in separate arrays and the boundary conditions are handled appropriately.

In *scheme 4* the momentum fluxes were modelled by means of the power-law method.[9] The acceleration within the control volume enclosing the nodal value ϕ_P was calculated using equation (12), but the influence coefficients were

$$a_E = \max(0, -F_e) + D_e \max(0, (1 - |F_e/D_e|)^5), \tag{18a}$$

$$a_W = \max(0, F_w) + D_w \max(0, (1 - |F_w/D_w|)^5), \tag{18b}$$

$$a_N = \max(0, -F_n) + D_n \max(0, (1 - |F_n/D_n|)^5), \tag{18c}$$

$$a_S = \max(0, F_s) + D_s \max(0, (1 - |F_s/D_s|)^5), \tag{18d}$$

$$a_P = a_E + a_W + a_N + a_S. \tag{18e}$$

The expressions for the influence coefficients a_E, a_W, a_N and a_S are similar to those used in connection with SIMPLE-type algorithms.[9,14]

In the present implementation, scheme 1 requires fewer arithmetic operations per time step than the other higher-order schemes. Also, the treatment of the mesh, free surface and internal obstacle boundaries is simpler than for the other higher-order schemes. However, this scheme does not ensure conservation of momentum in a discrete sense. Schemes 2 and 3 have the advantage that momentum conservation is ensured (except for round-off errors) while taking into account the effects of variable mesh spacing. Scheme 3 has the additional advantage of requiring fewer arithmetic operations per time step than scheme 2, but is formally less accurate. Scheme 4 was implemented mainly for comparison purposes, since it is formally less accurate and introduces a larger amount of numerical dissipation than the higher-order schemes. Nevertheless, this scheme is also of interest because it requires much less computational effort and fewer logical tests than the other schemes considered herein, since only five nodal values are required in the computation of $\partial\phi_P^n/\partial t$.

Once the acceleration $\partial\phi_P^n/\partial t$ computed using the old pressure gradient ∇p^n is available, the provisional velocity field $\mathbf{u}^* = (u^*, v^*)$ is calculated using the forward Euler method, i.e.

$$\phi_P^{n+1} = \phi_P^n + \Delta t\left(\frac{\partial\phi}{\partial t}\right)_P^n, \tag{19}$$

with $\phi^{n+1} = u^*$ or v^*. This provisional velocity field will not in general satisfy the continuity equation, because p^{n+1} is not yet available. Thus a second step is required to project $\mathbf{u}^* = (u^*, v^*)$ onto its divergence-free part, as will be described in the next section.

CONTINUITY EQUATION APPROXIMATIONS

To satisfy the continuity equation, a pressure correction must be determined so that the divergence of the temporary field is driven to zero. In the present work, two different schemes were used for solving the continuity equation: one based on the SOR method and the other on the conjugate residual method for solving systems of linear equations. The SOR method has the advantages of greater simplicity and easier treatment of the surface cells. However, its rate of convergence is poor, especially in problems involving large meshes. The conjugate residual method is more difficult to implement but has much greater computational efficiency.[3,15]

In the SOR implementation of the pressure equation the local pressure change is calculated from the condition $\nabla \cdot (\mathbf{u}^* + \delta\mathbf{u}) = 0$ in all full cells and from the condition $p^n + \delta p = 0$ in all surface cells.

In both cases the velocities located on the sides of the cell are simultaneously adjusted in response to the pressure change in the cell. For cells that are full of fluid the pressure correction is calculated from

$$\delta p_{\mathrm{P}} = -S_{\mathrm{P}}/(\partial S/\partial p)_{\mathrm{P}}, \tag{20}$$

with

$$S_{\mathrm{P}} = (u_{\mathrm{e}}^{*} - u_{\mathrm{w}}^{*})/\Delta x_i + (v_{\mathrm{n}}^{*} - v_{\mathrm{s}}^{*})/\Delta y_j \tag{21}$$

and

$$\beta = \frac{1}{(\partial S/\partial p)_{\mathrm{P}}} = \left\{ \frac{\Delta t}{\rho} \left[\frac{1}{\Delta x_i} \left(\frac{1}{\Delta x_{\mathrm{E}}} + \frac{1}{\Delta x_{\mathrm{W}}} \right) + \frac{1}{\Delta y_j} \left(\frac{1}{\Delta y_{\mathrm{N}}} + \frac{1}{\Delta y_{\mathrm{S}}} \right) \right] \right\}^{-1}. \tag{22}$$

The new iterates $(\;)^{(\nu)}$ for the pressure and the four cell face velocities are obtained from the previous iterates $(\;)^{(\nu-1)}$ by means of the equations

$$p_{\mathrm{P}}^{(\nu)} = p_{\mathrm{P}}^{(\nu-1)} + \delta p_{\mathrm{P}}, \tag{23a}$$

$$u_{\mathrm{e}}^{(\nu)} = u_{\mathrm{e}}^{(\nu-1)} + \Delta t \delta p_{\mathrm{P}}/(\rho \Delta x_{\mathrm{E}}), \tag{23b}$$

$$u_{\mathrm{w}}^{(\nu)} = u_{\mathrm{w}}^{(\nu-1)} - \Delta t \delta p_{\mathrm{P}}/(\rho \Delta x_{\mathrm{W}}), \tag{23c}$$

$$v_{\mathrm{n}}^{(\nu)} = v_{\mathrm{n}}^{(\nu-1)} + \Delta t \delta p_{\mathrm{P}}/(\rho \Delta y_{\mathrm{N}}), \tag{23d}$$

$$v_{\mathrm{s}}^{(\nu)} = v_{\mathrm{s}}^{(\nu-1)} - \Delta t \delta p_{\mathrm{P}}/(\rho \Delta y_{\mathrm{S}}). \tag{23e}$$

For cells that contain a free surface the pressure is computed in such a way that an interpolation between the pressure in the surface cell and the pressure in the nearest-neighbour full cell yields the correct pressure at the free surface. The pressure change can still be represented by equations (21) and (22), but with equation (21) replaced by $S_{\mathrm{P}} = (1 - \eta)p_{\mathrm{N}} + \eta p_{\mathrm{surf}} - p_{\mathrm{P}}$, where η is the ratio of the distance between the free surface and the centre of the interpolation neighbour cell and p_{surf} is the pressure at the free surface (assumed zero). One complete iteration consists of correcting the pressures and velocities in all cells occupied by fluid. The mesh must be swept several times until convergence is reached, i.e. until the velocity field obtained using equations (23a)–(23e) is discretely divergence-free.

The second procedure for finding the pressure correction, which for most problems is more efficient than the SOR method described above, was implemented as follows. By integrating equation (6) over the control volume surrounding the pressure node P and applying the Gauss theorem, the following equation is obtained:

$$a_{\mathrm{P}} \delta p_{\mathrm{P}} = a_{\mathrm{E}} \delta p_{\mathrm{E}} + a_{\mathrm{W}} \delta p_{\mathrm{W}} + a_{\mathrm{N}} \delta p_{\mathrm{N}} + a_{\mathrm{S}} \delta p_{\mathrm{S}} - \dot{m}, \tag{24}$$

where

$$a_{\mathrm{E}} = \Delta t \Delta y_j/(\rho \Delta x_{\mathrm{E}}), \tag{25a}$$

$$a_{\mathrm{W}} = \Delta t \Delta y_j/(\rho \Delta x_{\mathrm{W}}), \tag{25b}$$

$$a_{\mathrm{N}} = \Delta t \Delta x_i/(\rho \Delta y_{\mathrm{N}}), \tag{25c}$$

$$a_{\mathrm{S}} = \Delta t \Delta y_j/(\rho \Delta x_{\mathrm{S}}), \tag{25d}$$

$$a_{\mathrm{P}} = a_{\mathrm{E}} + a_{\mathrm{W}} + a_{\mathrm{N}} + a_{\mathrm{S}}, \tag{25e}$$

$$\dot{m}_{\mathrm{P}} = (u_{\mathrm{e}}^{*} - u_{\mathrm{w}}^{*})\Delta y_j + (v_{\mathrm{n}}^{*} - v_{\mathrm{s}}^{*})\Delta x_i. \tag{25f}$$

If the P-cell is an obstacle, void or surface cell, a_{E}, a_{W}, a_{N}, a_{S} and $\dot{m}_{\mathrm{P}}$ are set to zero and a_{P} is set to unity. If one of the neighbouring cells is a surface cell and the current cell is its interpolating neighbour, than it is necessary to account for the correct coupling between the two cells. For example, if the N-cell is a surface cell and the P-cell is the interpolating full cell, then δp_{N} is related to δp_{P} by

$$\delta p_{\mathrm{N}} = (1-\eta)\delta p_{\mathrm{P}} - p_{\mathrm{N}}^n + \eta p_{\mathrm{surf}} + (1-\eta)p_{\mathrm{P}}^n. \tag{26}$$

The appropriate coupling between the two cells can be achieved by setting $a_{\mathrm{N}} = 0$ and by adding $\eta\Delta t\Delta y_j/(\rho\Delta x_{\mathrm{E}})$ to a_{P} and $\eta\Delta t\Delta y_j/(\rho\Delta x_{\mathrm{E}})(p_{\mathrm{N}}^n - (1-\eta)p_{\mathrm{P}}^n - \eta p_{\mathrm{surf}})$ to $\dot{m}_{\mathrm{P}}$.

Equation (24) leads to a sparse symmetric system of linear equations in which the free surface, obstacle and perimeter boundary conditions are algebraically incorporated. Following Torrey's *et al.*,[3] this system is solved by the conjugate residual method. However, it is also possible to obtain the solution using line relaxation, the strongly implicit procedure (SIP) or a preconditioned conjugate gradient method.

The method described above is implemented in three steps. In the first step the coefficients and the source term $-\dot{m}_{\mathrm{P}}$ are computed for all cells in the mesh and the boundary conditions are incorporated. In the second step the pressure correction is computed by the conjugate residual method. In the final step, equations (7) and (8) are used to update the pressure and velocity fields so that $\mathbf{u}^{n+1} = (u^{n+1}, v^{n+1})$ is discretely divergence-free.

BOUNDARY CONDITIONS

In this section the formulation of the discretized boundary conditions will be described. At the regular mesh boundaries the velocity boundary conditions were imposed using a layer of fictitious cells surrounding the mesh, following the usual procedure. This technique has the advantage that the same expressions used for treating interior cells can also be used for cells adjacent to the mesh boundaries. Alternatively, if a conservative discretization method is used, the correct boundary conditions can be imposed by setting the corresponding influence coefficient to zero and then adding the contribution due to the boundary stress to the source term.

On the free-surface the simplified conditions $p = 0$ and $\mu(\partial u_\tau/\partial n) = 0$ were used, i.e. the effects of molecular viscosity and surface tension were neglected. Thus it was assumed that the fluid behaves like an ideal fluid in all surface cells.

The normal stress condition was imposed as a boundary condition for the pressure during the solution of the pressure correction equation. The tangential stress conditions were specified in the following way. First, the orientation of the free surface is sensed in a crude way.[2,3,10] For the cases in which the slope of the free surface was less than unity, it was assumed that $\partial/\partial n \approx \partial/\partial y$; otherwise the condition $\partial/\partial n \approx \partial/\partial x$ was assumed. The tangential boundary conditions were imposed by setting the velocities in the interface between surface and empty cells using the continuity equation and by setting the velocities in empty cells adjacent to partial fluid cells in such a way that $\partial u/\partial y = 0$ or $\partial v/\partial x = 0$. For example, in the situation depicted in Figure 4, these two conditions give

$$v_{\mathrm{n}} = v_{\mathrm{s}} - \frac{\Delta y_j}{\Delta x_i}(u_{\mathrm{ne}} - u_{\mathrm{e}})$$

and

$$u_{\mathrm{ne}} = u_{\mathrm{e}}$$

respectively. With this prescription, fictitious drag forces exerted on the fluid at surface cells are avoided.

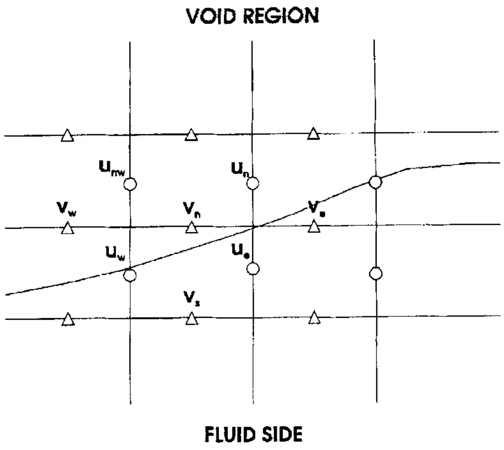


Figure 4. Definition sketch for implementation of numerical free surface boundary conditions

APPLICATIONS

Propagation of a solitary wave

The propagation of a solitary wave over a horizontal bottom is a classic test problem for free surface codes, because the problem is non-linear and the computed results can be compared with analytical solutions.

This application study was performed to evaluate the accuracy and stability properties of the discretizations described in the present work. The computed wave celerity and amplitude obtained using the four spatial schemes were compared with the theoretical solution for a highly non-linear solitary wave. The degradation of the wave profile due to numerical effects was evaluated by computing the RMS deviation between the computed and theoretical solutions.

The definition sketch for this test problem is presented in Figure 5. The space and time units were chosen so that $g = 1$ and $h = 1$. A solitary wave with dimensionless amplitude

$$\varepsilon_{\mathrm{w}} = \frac{H}{h} = 0{\cdot}5$$

and initial position $x_{\mathrm{crest}} = 10$ was defined in a computational domain 40 units long and 1·8 units high. The wave profile, fluid velocity and pressure were generated using the third-order theory developed

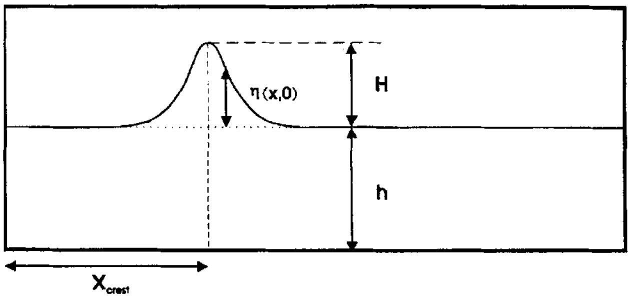


Figure 5. Definition sketch for solitary wave problem: H, wave height; h, still water level; η, free surface elevation; X_{crest}, initial position of wave crest

by Fenton.[16] In this approximation the wave celerity is

$$c = (gh)^{1/2}\left(1 + \frac{1}{2}\varepsilon_{\mathrm{w}} - \frac{3}{20}\varepsilon_{\mathrm{w}}^2 + \frac{3}{56}\varepsilon_{\mathrm{w}}^3\right) \tag{27}$$

and the elevation of the free surface above the still water level is obtained from the expression

$$\eta(x, t) = h\left[\varepsilon_{\mathrm{w}} S^2 - \frac{3}{4}\varepsilon_{\mathrm{w}}^2 S^2 T^2 + \varepsilon_{\mathrm{w}}^3\left(\frac{5}{8}S^2 T^2 - \frac{101}{80}S^4 T^2\right)\right], \tag{28}$$

where $S = 1/\cosh(x - ct)$ and $T = \tanh(x - ct)$. Although the solitary wave has infinite length, its effective length (where the momentum and energy are concentrated) is $\lambda_{\mathrm{eff}} = h/(3/4\varepsilon_{\mathrm{w}})^{1/2}$. Thus for the test problem $\lambda_{\mathrm{eff}} = 1{\cdot}6330$.

The computational domain was discretized using a mesh with $\bar{I} = 300$ cells in the x-direction with $\Delta x = 0{\cdot}133333$ (giving a numerical resolution of nearly 12 points per effective wavelength) and $\bar{J} = 24$ cells in the y-direction with $\Delta y = 0{\cdot}07$. The kinematic viscosity was specified as $v = 10^{-6}$ and the time step was $\Delta t = 0{\cdot}01$. Four numerical simulations were performed, using the same input data and varying the numerical scheme used in the momentum equations. The numerical simulations were stopped after $t = 15$ time units, which corresponds to 11·5 effective lengths of propagation.

The results of the calculations for the solitary wave problem at the end of the calculation time are summarized in Table I, which shows the computed amplitude and position of the wave crest for the theoretical solution and for the four numerical solutions obtained with schemes 1–4. Also included in this table is the mean quadratic deviation defined by the expression

$$\frac{1}{HN}\sum(\eta_{\mathrm{computed}}(x_i, t_{\mathrm{final}}) - \eta(x_0 - c_{\mathrm{computed}}t_{\mathrm{final}}))^2, \tag{29}$$

where η_{computed} is the wave elevation obtained in the numerical solution, t_{final} is the ending time and c_{computed} is the wave celerity obtained in the numerical solution. This quantity serves as an indicator of the degradation of the wave profile due to numerical effects. The results show that the non-conservative higher-order scheme produced more accurate predictions of the crest position (or wave celerity) and wave amplitude than the three conservative schemes. Also, the degradation of the wave profile due to numerical effects was smaller for scheme 1 than for the conservative discretizations. The numerical solutions obtained using schemes 2–4 were almost identical. Thus in this test problem the 'best' conservative scheme was scheme 4 (power-law discretization) because it required fewer arithmetic operations and logical tests to attain the same accuracy as the other conservative discretizations.

Figure 6 shows the evolution of the wave profiles for the computed solutions and the theoretical solution. The graphical output confirmed the numerical results presented in Table I. In particular, it is observed that scheme 1 reproduced the front profile of the solitary wave with remarkable accuracy. This figure shows that the degradation of the wave profile manifests itself in the appearance of small dispersive waves and in a slight forward tilting of the computed wave profiles. For all schemes the computed wave amplitude decreased at the beginning of the simulation (between $t = 0$ and 5) and then increased at a nearly constant rate. This shows that the numerical solutions were slightly unstable. The growing rate of the wave amplitude was slightly slower for the conservative schemes. However, it is apparent that the conservative schemes also induced larger dispersive losses than scheme 1. The fact that all schemes revealed a tendency to instability may be attributed to the very high Reynolds number used in the calculations and to the high dimensionless amplitude.

The fact that both low-order and higher-order schemes produced almost identical solutions is somewhat surprising. However, since the higher-order schemes are not used near the free surface, the

Table I. Computed wave crest position, wave height and mean quadratic deviation for theoretical solution and four numerical solutions obtained in solitary wave propagation test

	X_{crest}	Wave height	Mean quadratic deviation
Theoretical	34·3839	0·50000	—
Scheme 1	34·8730	0·46082	$5{\cdot}032 \times 10^{-4}$
Scheme 2	34·9298	0·47772	$8{\cdot}148 \times 10^{-4}$
Scheme 3	34·9401	0·47915	$7{\cdot}947 \times 10^{-4}$
Scheme 4	35·1075	0·49170	$8{\cdot}122 \times 10^{-4}$

performance of the higher-order schemes may have been affected by the small number of cells in the y-direction in the region above the still water level.

Numerical simulation of undular and breaking hydraulic jumps

The moving hydraulic jump is a well-studied type of flow[17–19] which allows simple and accurate checks of mass and momentum conservation in free surface flow computations. This type of flow can be generated using simple initial and boundary conditions. In the present work, moving hydraulic jumps were generated by pushing a uniform stream into a rigid vertical wall (see Figure 7).

If g, u_2 and h_2 are known, the discrete forms of the mass and momentum conservation laws can be combined into a cubic equation for the depth ratio $h^* = h_1/h_2$:

$$2h^*Fr = (1 + h^*)(1 - h^*)^2, \tag{30}$$

where $Fr = u_2^2/gh_2$ is the Froude number of the incident stream. For $Fr > 0$ this equation has two real positive roots and one real negative root. The physical solution corresponds to the positive root verifying the inequality $h^* > 1$.

In a Galilean reference frame moving with the jump celerity, the jump is seen as a sudden transition from subcritical to supercritical flow, characterized by an increase in water level and by a loss of energy of the nearly horizontal flow. According to the second law of thermodynamics, the resulting motion must be less organized than a nearly horizontal flow and thus may consist of either short waves or turbulence. It was found experimentally[20] that for $h^* > 1{\cdot}28$ the propagating jump maintains a purely undular character and the number of waves present at any instant is proportional to the time elapsed since the initiation of the jump. Lemoine[21] found the following approximate expressions for the amplitude a and wavelength λ of the jump waves:

$$\frac{a}{h_2} \approx \frac{1}{\sqrt{3}}\frac{h_2 - h_1}{h_1}, \tag{31}$$

$$\frac{\lambda}{h_2} \approx \frac{2\pi\sqrt{2}}{3}\left(\frac{h_2 - h_1}{h_1}\right)^{-1/2}. \tag{32}$$

Favre[20] verified experimentally that the wavelength of the jump waves is exceptionally sensitive to experimental conditions. Therefore we should expect that this wavelength should also be sensitive to the accuracy of the numerical scheme used to simulate jump waves. For $h^* > 1{\cdot}28$, breaking becomes increasingly prominent. For depth ratios greater than two the moving hydraulic jump is similar to a fully developed breaking wave. This process eventually leads to the production of turbulence, which dissipates energy from the incident stream.

In the present study, two moving hydraulic jumps were simulated. The first test case was a numerical simulation of Favre's experiment number 22. Abbott and Rodenhuis[22] computed the evolution of jump waves for this test case, using a numerical method based on the Boussinesq

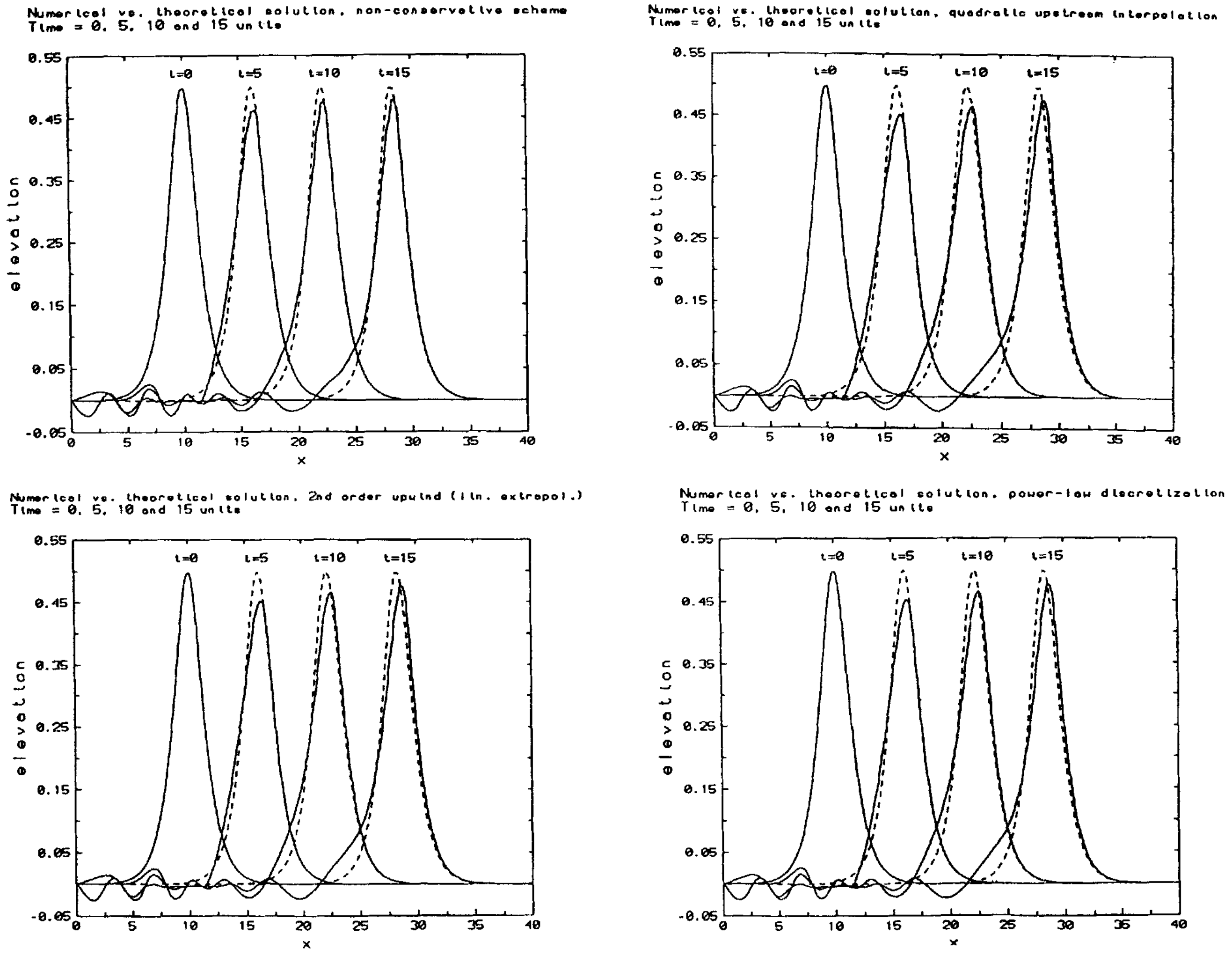


Figure 6. Comparison between theoretical and computed wave profiles for solitary wave test problem

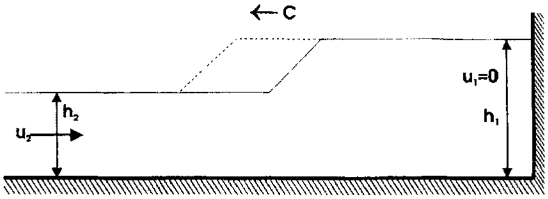


Figure 7. Definition sketch for moving hydraulic jump test case

equations for shallow water dispersive waves, and obtained computed wavelengths within 3 per cent of Favre's. The physical parameters used in the present study were chosen as close as possible to those of Favre's experiment. The computational domain was 6 m long and 0·2 m high, with a stream depth of 0·1075 m. The domain was discretized using a regular mesh of 200 × 20 cells with $\Delta x = 3$ cm and $\Delta y = 1$ cm. The moving jump was generated by imposing an initial horizontal velocity of 0·1386 m s^{-1}. The acceleration due to gravity was $g = 9{\cdot}81$ m s^{-2}. The molecular viscosity was chosen as $\nu = 1{\cdot}14 \times 10^{-6}$ m^2 s^{-1} and the time step was $\Delta t = 0{\cdot}005$ s. Four numerical experiments were performed, each with a different discretization scheme for the momentum equations. The simulations were stopped after $t = 4{\cdot}1$ s.

Figure 8 shows the evolution of the free surface configuration for the simulation performed with the quadratic upstream interpolation scheme. It is observed that the development of the jump waves was progressive. The height of the leading jump wave and the wavelength of the train increased during the simulation, but appeared to stabilize after an initial build-up.

The asymptotic wavelength λ_∞ and wave height a_∞ for the leading jump wave were estimated by means of an extrapolation procedure. The position and height of the two leading crests were plotted against $1/t$ and using a trial-and-error method it was found that the expressions

$$\lambda = \exp(a_1 + b_1/t + c_1/t^2), \qquad a = \exp(a_2 + b_2/t + c_2/t^2)$$

provided a good fit for the two relevant quantities. Thus $\lambda_\infty = \exp(a_1)$ and $a_\infty = \exp(a_2)$. Table II contains a summary of the results obtained in the four simulations of the undular jump.

It is observed that the wavelength λ_∞ is in fact slightly more sensitive to the numerical scheme used than the wave height a_∞. The wavelength determined experimentally by Favre for this test case was $\lambda_\infty = 1{\cdot}01$ m. The conservative schemes reproduced this value to three significant digits, whereas the non-conservative scheme predicted $\lambda_\infty = 0{\cdot}95$, which is the same value as predicted by the linear approximation of Lemoine. The non-conservative scheme predicted $a_\infty = 0{\cdot}133$ m, whereas all conservative schemes yielded $a_\infty = 0{\cdot}135$, which is the value determined experimentally by Favre. Thus it can be concluded that for this test case the conservative schemes produced better results than the higher-order non-conservative scheme. The computed fluid height at the wall after 4 s was 0·1222 m for the non-conservative scheme (scheme 1) and 0·1219 m for schemes 2–4, with errors of less than 1 per cent with respect to the theoretical value of 0·1225 m.

The second test case involving the propagation of hydraulic jumps was the simulation of a strong jump with depth ratio $h^* = 2$. In this case the rate at which energy is lost by the nearly horizontal flow cannot be compensated by the creation of dispersive waves. Thus the jump front becomes unstable and breaks and the energy is dissipated by large vortices and turbulence. The fluid height at the wall still provides a direct check of overall momentum conservation.

In this test case the space and time units were chosen so that $g = 1$ and $h = 1$. The input fluid velocity was $u = 0{\cdot}8860$ so that $h^* = 2$. The computational domain was 15 units long and 3·6 units high and was discretized using a mesh of 150 × 48 cells. The molecular viscosity was chosen as

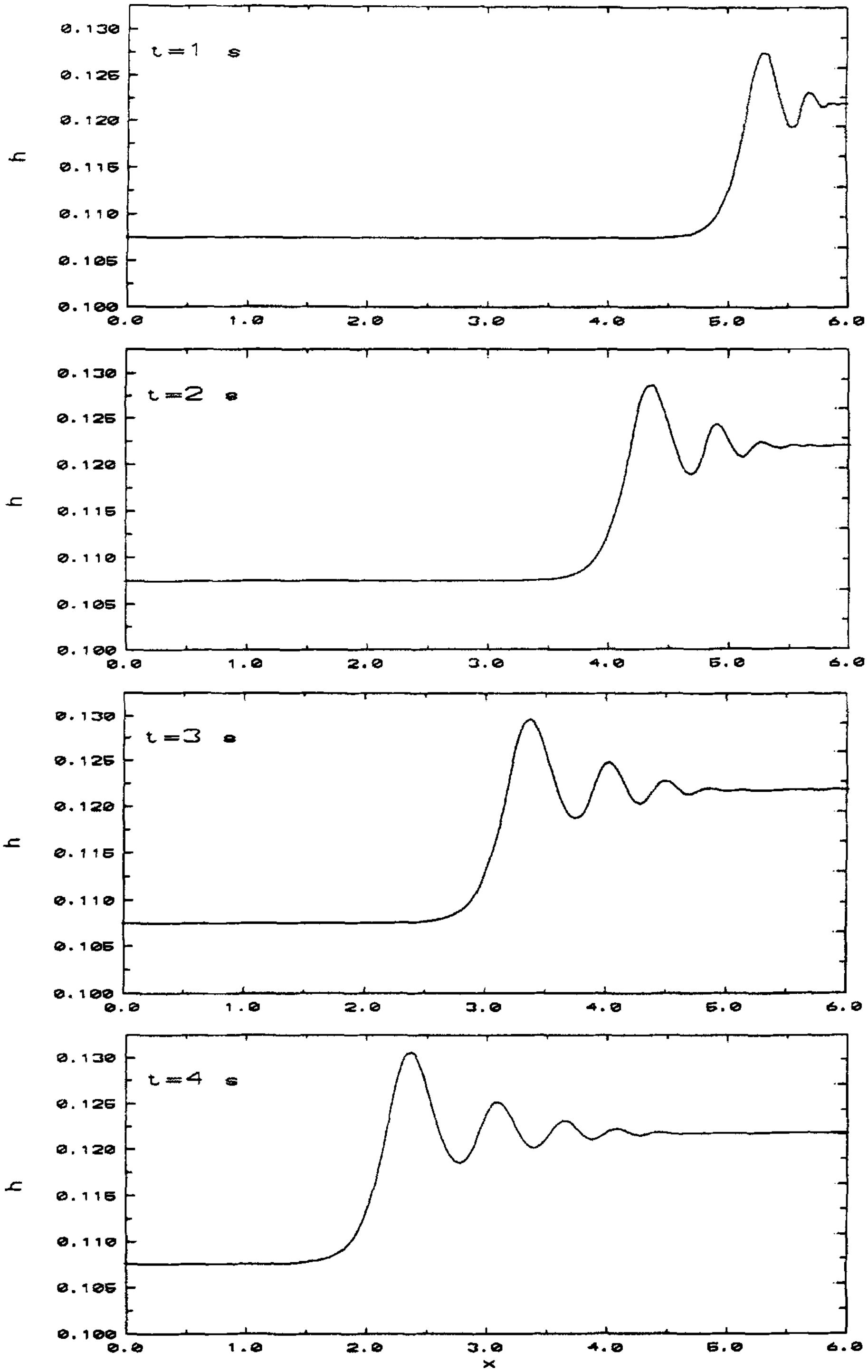


Figure 8. Development of jump waves, computed using quadratic upstream interpolation scheme

Table II. Extrapolated wavelength and wave height of leading jump wave obtained with four discretization schemes

	a (m)	a_∞ (m)	λ (m)	λ_∞ (m)
Scheme 1	$\exp(-2{\cdot}0204 - 0{\cdot}1120/t + 0{\cdot}0966/t^2)$	0·133	$\exp(-0{\cdot}0479 - 1{\cdot}3087/t + 0{\cdot}3948/t^2)$	0·953
Scheme 2	$\exp(-1{\cdot}9999 - 0{\cdot}1828/t + 0{\cdot}1659/t^2)$	0·135	$\exp(0{\cdot}0083 - 1{\cdot}4537/t + 0{\cdot}4952/t^2)$	1·008
Scheme 3	$\exp(-1{\cdot}9999 - 0{\cdot}1828/t + 0{\cdot}1659/t^2)$	0·135	$\exp(0{\cdot}0085 - 1{\cdot}4543/t + 0{\cdot}4956/t^2)$	1·09
Scheme 4	$\exp(-1{\cdot}9999 - 0{\cdot}1828/t + 0{\cdot}1659/t^2)$	0·135	$\exp(0{\cdot}0083 - 1{\cdot}4537/t + 0{\cdot}4952/t^2)$	1·008

$\nu = 10^{-6}$. Four numerical simulations were performed, each with a different discretization scheme. The computations were stopped after $t = 9{\cdot}1$ units.

Figure 9 shows three velocity and free surface configuration plots obtained using scheme 2. These plots describe the initial stage of breaking, when the free surface folds onto itself, and a subsequent cycle of splashing and vortex formation. The large-scale 'roller-type' vortices were generated by the break-up of the surface shear layer created by the plunging front. The plot corresponding to $t = 9{\cdot}0$ shows two well-developed vortices, with a third one starting to form near the toe of the jump. The

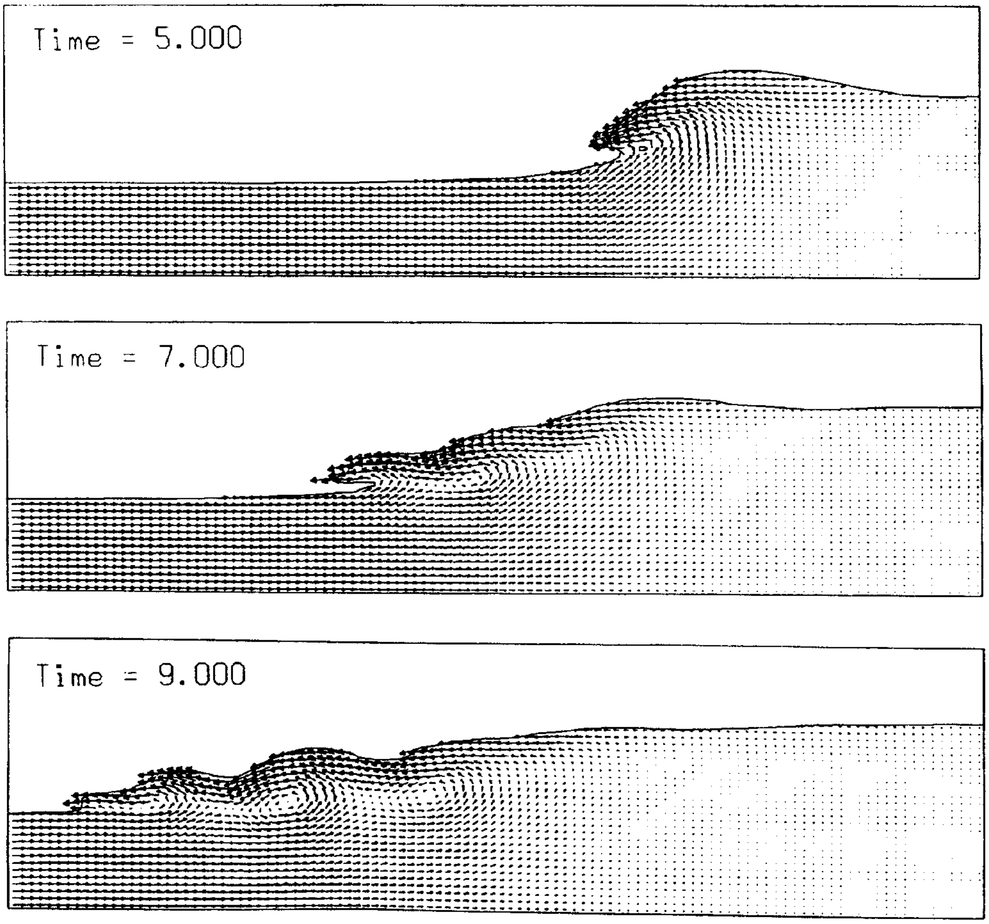


Figure 9. Development of breaking hydraulic jump, computed using quadratic upstream interpolation scheme

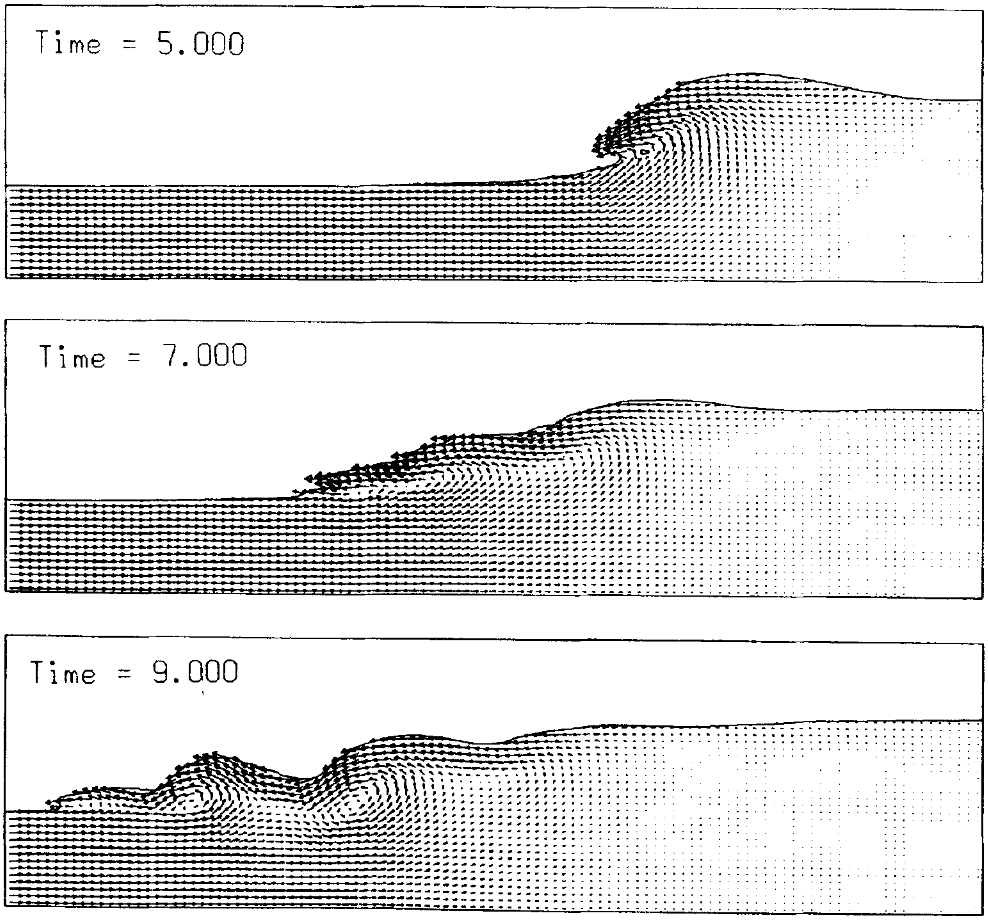
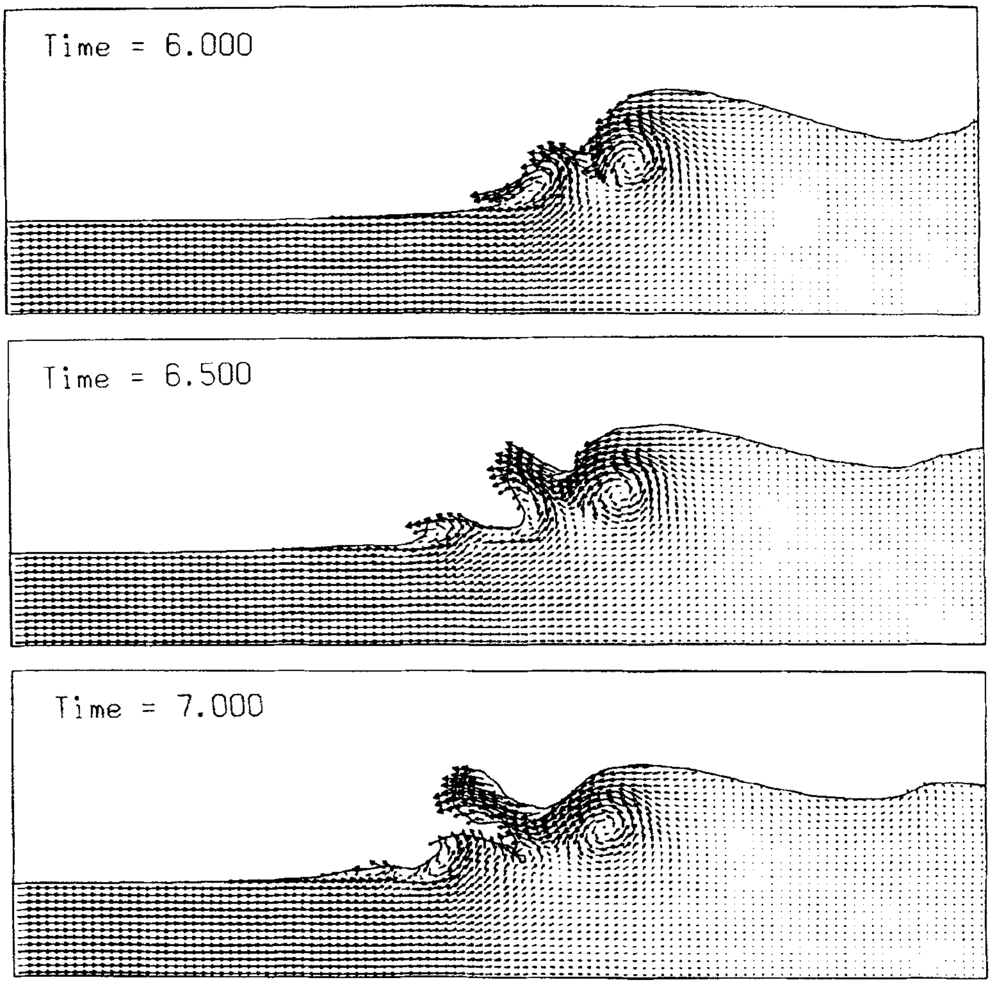


Figure 10. Development of breaking hydraulic jump, computed using power-law discretization scheme

results obtained using scheme 3 were similar to those obtained using scheme 2, but with a slight smearing of the vortices and of the fine structures of the flow (e.g. the corrugations of the free surface during the splashing cycle).

Figure 10 shows three velocity and free surface configuration plots obtained using the lower-order power-law scheme. The qualitative characteristics of the solution were similar to those of the solution obtained using scheme 2, but some differences are also apparent. First, there was a slight smearing of the fine structures during the initial splash. This is apparent in the plot corresponding to $t = 7{\cdot}0$. Second, the vortex structures away from the advancing front appeared to dissipate more quickly than in the simulation performed with scheme 2. Finally, the vortices were elongated in a slighly different spatial orientation.

The numerical simulation performed using scheme 1 became unstable just after the initiation of breaking. Figure 11 illustrates three velocity and free surface plots obtained using scheme 1, but with the viscosity increased to $\nu = 10^{-5}$. It is observed that the computed solution was unstable, with non-physical features developing rapidly. Another indication of the poor performance of scheme 1 was the fluid height at the wall. The average fluid height at the wall after the initial upsurge was 1·95 for scheme 1, 2·03 for schemes 2 and 3 and 2·02 for scheme 4.



Time = 6.000

Time = 6.500

Time = 7.000

Figure 11. Development of breaking hydraulic jump, computed using non-conservative scheme

Thus it can be concluded that in this test case, which involves recirculation and breaking, discrete momentum conservation is more important for the quality of the numerical solution than the order of the scheme used for the spatial discretization.

CONCLUSIONS

In the present work, four spatial discretization schemes for the integration of the momentum equation were incorporated in a computer code for free surface flows with arbitrary configurations based on the VOF method. The purpose of the work was to evaluate the influence of formal accuracy and discrete momentum conservation on the stability and accuracy of the numerical solutions for problems involving large free surface distortions.

The numerical schemes considered in the present work were (i) a non-conservative scheme based on four-point approximations for the spatial derivatives in the advection term, (ii) a conservative scheme based on quadratic upstream interpolation, (iii) a conservative scheme based on linear extrapolation and (iv) a lower-order conservative scheme based on the power-law discretization. These four schemes were compared in three test problems: the propagation of a high-amplitude solitary wave, the propagation of an undular hydraulic jump and the propagation of a breaking hydraulic jump.

The results obtained for the solitary wave problem showed that the four schemes reproduced accurately the theoretical propagation velocity, even for a relatively coarse discretization. However, it was found that the non-conservative scheme was potentially unstable. The simulation of the moving undular jump showed that the conservative schemes performed slightly better than the non-conservative scheme. Finally, the simulation of the breaking hydraulic jump showed that the non-conservative scheme produced unstable or physically incorrect solutions, whereas the three conservative schemes yielded good numerical solutions. The lower-order scheme introduced a slight smearing of the fine details of the flow in regions containing thin laminae of fluid or where the velocity gradient was small.

The numerical solutions obtained in the present work showed that in free surface flow computations, discrete momentum conservation is more important than the formal accuracy of the discretization. Among the higher-order schemes, the quadratic upstream interpolation scheme (scheme 2) performed slightly better than the linear extrapolation scheme (scheme 3). It was found that the lower-order scheme (scheme 4) performed almost as well as the higher-order conservative schemes. It is expecteed that the difference between the various conservative schemes can only be established by improving other characteristics of the numerical algorithm (time-stepping scheme, numerical boundary conditions, transport algorithm for the F-function, etc.) or by considering more difficult test cases.

APPENDIX: NOMENCLATURE

a_E, a_W, a_N, a_S, a_P	coefficients of finite difference equations
D_w, D_e, D_n, D_s	diffusion coefficients for faces w, e, n, s of P-control volume
$F(\mathbf{x}, t)$	volume-of-fluid (VOF) function
Fr	Froude number in hydraulic jump problem
F_w, F_e, F_n, F_s	mass flow rates through faces w, e, n, s of P-control volume
g	gravitational acceleration
$\mathbf{g}$	(g_x, g_y), vector of body force acceleration
g_x, g_y	components of body force acceleration in x- and y-directions
h	still water level in solitary wave problem
h^*	depth ratio for hydraulic jump
H	wave height of solitary wave
J_w, J_e, J_n, J_s	momentum fluxes through faces w, e, n, s of P-control volume
$\dot{m}$	residual divergence in pressure correction equation
n	direction normal to boundary
p	pressure
S_p	residual function in pressure correction equation
t	time
u	velocity component in x-direction
$\mathbf{u}$	(u, v) velocity vector
$\mathbf{u}_b$	boundary velocity
u^*	temporary velocity component in x-direction
$\mathbf{u}^*$	temporary velocity
v	velocity component in y-direction

v^*	temporary velocity component in y-direction
Vol	volume of P-control volume
x	horizontal co-ordinate
$\mathbf{x}$	(x, y), position vector
y	vertical co-ordinate

Greek letters

Γ_{D}	part of boundary on which Dirichlet boundary conditions are applied
Γ_{N}	part of boundary on which Neumann boundary conditions are applied
Δ	space or time increment
μ	coefficient of dynamic viscosity
ν	coefficient of kinematic viscosity
ρ	density (uniform in present work)
ϕ	generic variable in advection–diffusion equation (u or v in present work)

REFERENCES

1. F. H. Harlow and J. E. Welch, 'Numerical calculation of time-dependent viscous incompressible flow', *Phys. Fluids*, **8**, 2182–2189 (1965).
2. B. D. Nichols and C. W. Hirt, 'Methods for calculating multi- dimensional, transient free-surface flows past bodies', *Proc. 1st Int. Conf. on Numerical Ship Hydrodynamics*, Gaithersburg, MD, October 1975.
3. M. D. Torrey, L. D. Cloutman, R. C. Mjolsness and C. W. Hirt, 'NASA-VOF2D: a computer program for incompressible flows with free surfaces', *Los Alamos National Laboratory Rep. LA-10612-MS*, 1985.
4. C. W. Hirt and J. P. Shannon, 'Free-surface stress conditions for incompressible-flow calculations', *J. Comput. Phys.*, **2**, 403–411 (1968).
5. B. D. Nichols and C. W. Hirt, 'Improved free-surface boundary conditions for numerical incompressible flow calculations', *J. Comput. Phys.*, **8**, 434–448 (1971).
6. C. M. Lemos, 'A simple numerical technique for turbulent flows with free surfaces', *Int. j. numer. methods fluids*, **15**, 127–146 (1992).
7. R. W. Davies and E. F. Moore, 'Numerical study of vortex shedding from rectangles', *J. Fluid Mech.*, **116**, 475–506 (1982).
8. S. S. Thakur and W. Shyy, 'Some implementational issues of convection schemes for finite volume formulations', *Numer. Heat Transfer B*, **24**, 31–55 (1993).
9. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1980.
10. B. D. Nichols, C. W. Hirt and R. S. Hotchkiss, 'SOLA-VOF: a solution algorithm for transient fluid flow with multiple free-boundaries', *Los Alamos Scientific Laboratory Rep. LA-8355*, 1980.
11. P. Roache, *Computational Fluid Dynamics*, Hermosa, Albuquerque, NM, 1972.
12. P. M. Gresho, 'Some current CFD issues relevant to the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **87**, 201–252 (1991).
13. B. P. Leonard, 'A stable and accurate convective modelling procedure based on quadratic upstream interpolation', *Comput. Methods Appl. Mech. Eng.*, **19**, 59–98 (1979).
14. C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics*, Springer, Berlin, 1988.
15. R. Chandra, 'Conjugate gradient methods for partial differential equations', *Ph.D. Thesis*, Computer Science, Yale University, New Haven, CT, 1978.
16. J. Fenton, 'A ninth-order solution for the solitary wave', *J. Fluid Mech.*, **53**, 257–271 (1972).
17. N. Rajaratnam, 'Hydraulic jumps', *Adv. Hydrosci.*, **4**, 197–280 (1967).
18. F. J. Resh and H. J. Leutheusser, 'Reynolds stress measurements in hydraulic jumps', *J. Hydraul. Res.*, **10**, 409–430 (1972).
19. P. A. Madsen and I. A. Svendsen, 'Turbulent bores and hydraulic jumps', *J. Fluid Mech.*, **129**, 1–25 (1983).
20. H. Favre, *Ondes de Translation dans les Canaux Découverts*, Dunod, Paris, 1935.
21. R. Lemoine, 'Sur les ondes positives de translation dans les canaux et sur le ressault ondulé de faible amplitude', *La Houille Blanche*, **2**, (1948).
22. M. B. Abbott and C. S. Rodenhuis, 'A numerical simulation of the undular hydraulic jump', *J. Hydraul. Res.*, **12**, 141–152 (1972).