Eur. J. Mech. B - Fluids 20 (2001) 57–74

S0997-7546(00)01104-3/FLA

Solitary wave breaking on sloping beaches: 2-D two phase flow numerical simulation by SL-VOF method

Stephan Guignard [a,b], Richard Marcer [b], Vincent Rey [a], Christian Kharif [c], Philippe Fraunié [a,*]

[a] *LSEET, UMR 6017 du CNRS, Université de Toulon et du Var, BP 132, 83957 La Garde cedex, France*
[b] *Principia R&D S.A., Z.I. Bregaillon, 83507 La Seyne-sur-mer cedex, France*
[c] *Ecole Supérieure de Mécanique de Marseille, IMT-Technopôle de Château-Gombert, 13451 Marseille cedex 20, France*

(Received 11 January 1999; revised 21 December 1999; accepted 4 July 2000)

Abstract – This paper describes the development of a computational method for simulating breaking and post-breaking of solitary waves over sloping bottoms. The Navier–Stokes equations are solved in air and water with respect to the real density ratio between the two fluids using a pseudo-compressibility method. The interface tracking is achieved by a new method, called Segment Lagrangian-Volume Of Fluid (SL-VOF), using the well known concepts of VOF, Piecewise Linear Interface Calculation (PLIC) and adding Lagrangian advection of the segments representing the interface. The verification of this method is made through various simple test cases. Results concerning wave shoaling are compared with those of Boundary Integral Element Method (BIEM) simulations, that are known to be very accurate up to breaking. The SL-VOF method is able to simulate the flow beyond the point at which the interface impacts on itself. Simulations of the breaking stage are compared with experiments. In both cases a very good agreement is observed. © 2001 Éditions scientifiques et médicales Elsevier SAS

solitary wave / wave breaking / Volume Of Fluid

1. Introduction

In coastal zones, shoreward wave propagation generally leads to breaking. This phenomenon induces sediments suspension, energy dissipation and momentum exchange between waves and nearshore wave-induced currents. Such long-shore currents, rip currents, or undertow may carry sediments away and change the local bathymetry. The beach profile evolution is not the only effect of breaking waves: as they are the most violent waves, they can cause serious damage on offshore and environmental protective structures. Wave breaking is a highly non-linear phenomenon involving two turbulent flows of very different densities. This is the reason why measurements of the dynamics of breaking waves are very difficult and why simple theories or numerical models are unable to deal with such problems. Higher-order models, able to treat strongly non-linear interfaces between fluids are required for describing such a phenomenon. Thus, since the seventies, many authors have been developing numerical methods able to deal with wave breaking. The first accurate numerical method used to simulate this extreme event was the boundary integral element method (BIEM) of Longuet-Higgins and Cokelet [1]. This method, which solves the potential flow theory equations was later successfully used by several authors: e.g. Peregrine [2] and Skyner [3] for plunging breakers in deep water, Grilli et al. [4] for solitary waves breaking on sloping beaches, Yasuda et al. [5,6] for reefs and Grilli et al. [7] for submerged breakwaters. Limitations of these very accurate and fast methods result from the physical hypotheses of irrotational motion of an inviscid fluid and neglecting the effect of air, on the one hand and from

* Correspondence and reprints.
E-mail address: fraunie@lseet.univ-tln.fr (P. Fraunié).

the requirement of a simply-connected computational domain on the other hand. Hence, breaking events cannot be completely described by such algorithms: after a breaker jet impacts on the free surface, a rotational viscous two phase flow must be modelled.

Interface traking methods have been reviewed by Hyman [8]. Recently, Monaghan et al. [9] applied the Smoothed Particle Hydrodynamics method (SPH) to wave breaking induced by a moving wall, but only a few results were given.

The Volume Of Fluid (VOF) method has been introduced by Hirt and Nichols [10] through the SOLAVOF algorithm. In the VOF method, the interface is represented by the fraction of the grid cell occupied by the denser fluid. Whereas VOF methods are numerically less accurate than the BIEM methods, they are able to deal with several interacting interfaces. Moreover, the accuracy of the VOF method has been constantly improved over the last decade. In the FLAIR method (Flux Line Segment Model for Advection and Interface Reconstruction), the interface is represented by segments that are not necessarily parallel to the grid faces. Lafaurie et al. [11] presented a new numerical method based on the VOF method of Hirt and Nichols to simulate the behavior of a mixture of immiscible fluids, when surface tension must be taken into account. Li [12] proposed the PLIC (Peacewise Linear Interface Calculation) which is a second-order interface tracking method.

Regarding the application of VOF methods to wave breaking, Petit et al. [13] applied FLAIR to the modeling of periodic waves travelling shoreward and breaking over dykes. They solved the Navier–Stokes equations in water but did not give an accurate description of the interface shape evolution during breaking. The PLIC method was applied by Abadie [14] to the case of wave breaking in the waters of finite depth, taking into account the air flow. Chen et al. [15] performed direct numerical simulations with PLIC of the two-phase flow due to the breaking of short gravity waves, including surface tension effects. They solved the Navier–Stokes equations in the whole domain. The two latter studies give a detailed description of plunging breaker dynamics. In these numerical investigations, wave breaking is not due to the bathymetric effects but is forced by using a theoretical steadily propagating wave of steepness higher than the maximum admissible value. These waves are known to break rapidly. Recently, Lin and Liu [16,17] performed numerical simulations of breaking waves in the surf zone, using the SOLAVOF algorithm and including a $(\kappa - \varepsilon)$ turbulence model. Interesting results, obtained concerning turbulence under breaking waves are given, but no details of the interface evolution during breaking were shown.

The goal of the present study is to simulate the wave breaking due to spatial bottom variations, taking into account viscous and rotational effects as well as the interaction between two fluids of very different densities. The Navier–Stokes equations are solved using the pseudo-compressibility method (Viviand) [18] and the interface treatment is based on a new interface traking method referred to as SL-VOF. The case of a solitary wave shoaling over a plane slope is presented as an application of the method. Results concerning the interface shape computed in this case are compared with BIEM results up to the jet impact and with experimental results obtained during the breaking event.

2. The numerical problem

2.1. Formulation

The problem consists in solving the mass and momentum balance equations for two incompressible viscous fluids of different densities separated by a moving interface, in a two-dimensional vertical flow (2DV). In a Cartesian reference system xOy, Oy being vertical upward, these balance equations can be written:

$$\frac{\partial W}{\partial t} + \frac{\partial F_0}{\partial x} + \frac{\partial G_0}{\partial y} = R \tag{1}$$

with

$$W = \begin{bmatrix} 0 \\ \rho u \\ \rho v \end{bmatrix}; \qquad F_0 = \begin{bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \end{bmatrix}; \qquad G_0 = \begin{bmatrix} \rho v \\ \rho uv - \tau_{yx} \\ \rho v^2 + p - \tau_{yy} \end{bmatrix}; \qquad R = \begin{bmatrix} 0 \\ 0 \\ -\rho g \end{bmatrix}, \tag{2}$$

where u and v are the velocity field components, p the pressure field, τ_{ij} $(i, j = x, y)$ the viscous stress tensor components, ρ the density of each fluid and g the gravitational acceleration. The computational grid is curvilinear, in order to allow accurate discretization schemes, a coordinate transformation is used so that the original mesh is changed into a square orthogonal one. The coordinate transformation being defined by $(x_1(x, y) = cte;\ y_1(x, y) = cte)$, equation (1) can be re-written:

$$\frac{\partial}{\partial t}\left(\frac{W}{J}\right) + \frac{\partial F_1}{\partial x_1} + \frac{\partial G_1}{\partial y_1} = \frac{R}{J} \tag{3}$$

with

$$\begin{bmatrix} F_1 \\ G_1 \end{bmatrix} = \begin{bmatrix} M \\ J \end{bmatrix} \begin{bmatrix} F_0 \\ G_0 \end{bmatrix} \tag{4}$$

and

$$M = \begin{bmatrix} \dfrac{\partial x_1}{\partial x} & \dfrac{\partial x_1}{\partial y} \\ \dfrac{\partial y_1}{\partial x} & \dfrac{\partial y_1}{\partial y} \end{bmatrix}; \qquad J = DET(M). \tag{5}$$

2.2. The pseudo-compressibility method

2.2.1. Principle of the method

Equation (3) is replaced by a new set of equations, discretized in time as follows:

$$\begin{aligned} &\frac{\partial}{\partial \tau}\left(\frac{\tilde{W}}{J}\right)^{n+1} + \frac{1}{2\Delta t}\left[3\left(\frac{W}{J}\right)^{n+1} - 4\left(\frac{W}{J}\right)^{n} + \left(\frac{W}{J}\right)^{n-1}\right] + \left(\frac{\partial F_1}{\partial x_1}\right)^{n+1} + \left(\frac{\partial G_1}{\partial y_1}\right)^{n+1} = \left(\frac{R}{J}\right)^{n+1}, \\ &p^{n+1} = \rho\left(U_0^2 + \chi \vec{u}_n^2\right) \ln\left(\frac{\tilde{\rho}}{\rho}\right)^{n+1} + cte \end{aligned} \tag{6}$$

with

$$\tilde{W} = \begin{bmatrix} \tilde{\rho} \\ \tilde{\rho} u \\ \tilde{\rho} v \end{bmatrix} = \frac{\tilde{\rho}}{\rho} W. \tag{7}$$

$\tilde{\rho}$ is the pseudo-density and τ the pseudo-time variable. An appropriate number of iterations in τ between two iterations in t lets the solution of this system tend to the solution of the unsteady system. The spatial discretization is centered (see section 2.3.3). In order to avoid uncoupling between odd and even (i, j) cells, it

appears necessary to use artificial dissipation. So the first equation of the system (6) is written:

$$\frac{\partial}{\partial\tau}\left(\frac{\tilde{W}}{J}\right)^{n+1} + Q(\tilde{W}) = 0, \tag{8}$$

where $Q(\tilde{W})$ groups the flux terms, the unsteady terms and the source terms.

By adding the artificial dissipation term $D(\tilde{W})$ (Viviand [18]), equation (8) becomes:

$$\frac{\partial}{\partial\tau}\left(\frac{\tilde{W}}{J}\right)^{n+1} + Q(\tilde{W}) - D(\tilde{W}) = 0. \tag{9}$$

2.2.2. Time marching

Discretization of the pseudo-time is made as follows, through a five stage Runge–Kutta (R-K) scheme (m being the index of the pseudo-time steps and q the index of the R-K scheme step, $q = 1, \ldots, 5$):

$$\frac{1}{\Omega_q \Delta\tau}\left[\left(\frac{\tilde{W}}{J}\right)^{m+q/5} - \left(\frac{\tilde{W}}{J}\right)^{m}\right] + \frac{3}{2\Delta t}\left(\frac{W}{J}\right)^{m+q/5} = L^{m+(q-1)/5}, \tag{10}$$

where

$$L^{m+(q-1)/5} = -\left(\frac{\partial F_1}{\partial x_1} + \frac{\partial G_1}{\partial y_1}\right)^{m+(q-1)/5} + \left(\frac{R}{J}\right)^{n+1} - \frac{1}{2\Delta t}\left[-4\left(\frac{W}{J}\right)^{n} + \left(\frac{W}{J}\right)^{n-1}\right] + D(\tilde{W})^{m+(q-1)/5}$$

and

$$\Omega_1 = \frac{1}{4}, \qquad \Omega_2 = \frac{1}{6}, \qquad \Omega_3 = \frac{3}{8}, \qquad \Omega_4 = \frac{1}{2}, \qquad \Omega_5 = 1.$$

The first equation of the system (10) gives $\tilde{\rho}^{m+q/5}$ and, since $W = \rho/\tilde{\rho}\,\tilde{W}$, the second and third equations of (10) become:

$$\left[\frac{1}{\Omega_q \Delta\tau} + \frac{3}{2\Delta t}\frac{\rho}{\tilde{\rho}^{m+q/5}}\right]\left(\frac{\tilde{W}_k}{J}\right)^{m+q/5} = \frac{1}{\Omega_q \Delta\tau}\left(\frac{\tilde{W}_k}{J}\right)^{m} + L^{m+(q-1)/5} \tag{11}$$

for $k = 2, 3$.

2.2.3. Pseudo-time step

The value of the pseudo-time step $\Delta\tau$ is obtained from a stability criterion (Viviand [18]; Laget [19,20]). Its local value is computed as follows:

$$\Delta\tau_{i,j} = \Delta\tau^0_{i,j}\frac{1}{1 - R_{i,j}} \quad \text{with } R_{i,j} = \frac{3\tilde{\rho}^{m+(q-1)/5}_{i,j}\Delta\tau^0_{i,j}}{2\,\rho^n_{i,j}\Delta t\,\mathrm{CFL}}, \tag{12}$$

where $\Delta\tau^0_{i,j}$ is the local pseudo-time step computed for the stationary problem (Vatsa [21]), and CFL is a constant that has to be adjusted by the user in order to achieve stability and local convergence.

The value of $\Delta\tau^0_{i,j}$ is computed with respect to a stability criterion combining inviscid fluid behavior and viscous effects:

$$\Delta\tau^0_{i,j} = \frac{\mathrm{CFL}}{\lambda_{x(i,j)} + \lambda_{y(i,j)} + (\nu_{x(i,j)} + \nu_{y(i,j)})(8\mu_{i,j}/3\tilde{\rho}^{m+(q-1)/5}_{i,j})}, \tag{13}$$

where

$$\nu_{x(i,j)} = \left(\frac{\partial x_1}{\partial x}\right)^2 + \left(\frac{\partial x_1}{\partial y}\right)^2; \qquad \nu_{y(i,j)} = \left(\frac{\partial y_1}{\partial x}\right)^2 + \left(\frac{\partial y_1}{\partial y}\right)^2,$$

$\mu_{i,j}$ is the local viscosity.

$\lambda_x = \max(|\alpha_x|, \frac{1}{2}|\alpha_x|\sqrt{\beta_x})$ is the maximum eigenvalue of the Jacobian flux matrix in the direction x_1, with:

$$\alpha_x = \frac{\rho}{\tilde{\rho}} u_1,$$

$$\beta_x = \alpha_x\alpha_x + 4\frac{\rho_{i,j}\nu_{x(i,j)}}{\tilde{\rho}\frac{\partial\tilde{\rho}}{\partial p}}.$$

λ_y is defined in the same way by replacing x by y and u_1 by v_1, given by:

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} M \\ J \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}. \tag{14}$$

2.2.4. Spatial discretization

• Advection, pressure and viscous terms: F_1, G_1. The formulation is a finite volume one and the discretization of these terms is the following one:

$$F_{1,i+\frac{1}{2},j} = \frac{1}{2}(F_{1,i,j} + F_{1,i+1,j});$$

$$G_{1,i,j+\frac{1}{2}} = \frac{1}{2}(G_{1,i,j} + G_{1,i,j+1}).$$

• Artificial dissipation terms:

The formulation here follows Jameson and Baker [22], that is, in the x_1 direction:

$$D_{i+\frac{1}{2},j} = \lambda_{i+\frac{1}{2},j}\left[\varepsilon^{(2)}_{i+\frac{1}{2},j}(\tilde{W}_{i+\frac{1}{2},j} - \tilde{W}_{i,j}) + \varepsilon^{(4)}_{i+\frac{1}{2},j}(\tilde{W}_{i+2,j} - 3\tilde{W}_{i+1,j} + 3\tilde{W}_{i,j} - \tilde{W}_{i-1,j})\right];$$

where

$$\varepsilon^{(2)}_{i+\frac{1}{2},j} = vis_2 \cdot \max(\omega_{i+1}, \omega_i);$$

$$\omega_i = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{p_{i+1,j} + 2p_{i,j} + p_{i-1,j}};$$

$$\varepsilon^{(4)}_{i+\frac{1}{2},j} = \max\left[0\left(vis_4 - \varepsilon^{(2)}_{i+\frac{1}{2},j}\right)\right];$$

vis_2 and vis_4 are fixed by the user, generally equal to 0 or 1.

$\lambda_{i+\frac{1}{2},j}$ is computed according to Vatsa's [21] anisotropic model:

$$\lambda_{i+\frac{1}{2},j} = \min(\lambda_{i,j}, \lambda_{i+1,j}).$$

2.3. Interface treatment

The interface and its displacement are obtained for each time step of the simulation by using a new method based on the two well known concepts of VOF (Hirt and Nichols [10]) and PLIC (Li [12]).

2.3.1. Interface modelling

The interface is calculated in each cell using a discrete function C whose value in each cell is the fraction of the cell occupied by the denser fluid (VOF concept). The original SOLAVOF (Hirt and Nichols [10]) method assumes the interface to be parallel to the grid faces, whereas the PLIC method allows the interface to be represented by segments of any orientation. As shown in *figure 1*, the normal $\vec{n}$ to the interface in each cell is defined as $-\vec{\nabla}(C)$. Thus, it is possible to represent the interface in each cell by a segment set normal to $\vec{n}$, whose position is determined so that the fraction in the corresponding cell is equal to the value of C in this cell (PLIC concept). Note that this representation does not ensure a continuous representation of the interface: a continuous piecewise linear interface modelling would not satisfy the cells volume fraction.

As proposed by Li [12], the coordinates of $\vec{n}$ are computed using a eight points discretization:

$$n_x = \frac{1}{8}\big((C_{i+1,j+1} - C_{i-1,j+1}) + 2(C_{i+1,j} - C_{i-1,j}) + (C_{i+1,j-1} - C_{i-1,j-1})\big), \tag{15}$$

$$n_y = \frac{1}{8}\big((C_{i-1,j+1} + 2C_{i,j+1} + C_{i+1,j+1}) - (C_{i-1,j-1} + 2C_{i,j-1} + C_{i+1,j-1})\big). \tag{16}$$

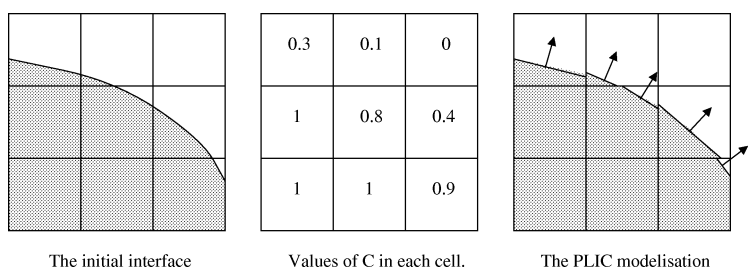

Figure 1. PLIC modelling of the interface.

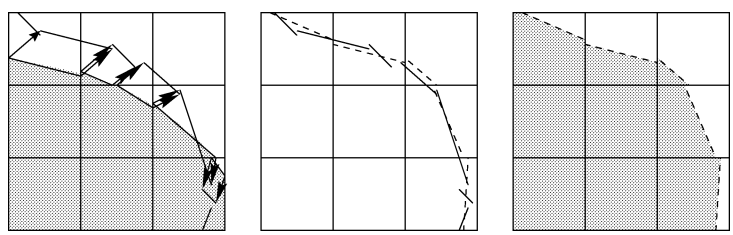

Figure 2. SL-VOF general principle, left: interface advection; middle: advected (-) and reconstructed (–) segments; right: recalculated C^{n+1} field.

2.3.2. Interface advection

The velocity at the ends of each segment is deduced from the general velocity field by a bilinear interpolation. The ends of the segments are then advected using a first-order Lagrangian scheme (*figure 2*, left square, the black arrows represent the velocity vectors):

$$\vec{x}(t+\delta t) = \vec{x}(t) + \vec{u}\,\delta t. \tag{17}$$

2.3.3. Computation of C^{n+1}

The PLIC method uses a multi-step flux calculation to compute the value of C^{n+1}. Here instead, values of C^{n+1} are deduced from the new positions of the segments after the interface advection.

For this purpose, markers are uniformly spaced on each segment. To each marker M_i is associated the normal vector $\vec{n}_i$ of the segment to which it belongs (*figure 3*). Two types of cells are then to be considered: type A corresponds to the cells containing at least one marker and type B corresponds to cells containing no marker.

Let us consider a type A cell. To the marker M_i corresponds a value C_i calculated according to the PLIC concept as shown in *figure 3*. A test has to be made here on the value of $pr_{ij} = \vec{n}_i \cdot \vec{n}_j$, i and j describing the markers of the cell. If one of the pr_{ij} is negative, we are in the presence of a reconnection scenario: i.e. areas containing the heavier fluid join to fill-up this cell, as shown in *figure 4*, the value of the C function in the cell is set equal to one. If none of the $pr_{i,j}$ is negative, the C value will be set to the mean value of the C_i's in the cell.

Let us now consider a type B cell: (i) if the cell did not contain a segment before the advection of the interface, the value of C for this cell is not modified; (ii) if the cell did contain a segment before advection, one has to detect whether the cell becomes empty or full of the denser fluid, after advection. This is achieved here by considering the values of $pr = \vec{n} \cdot \vec{d}$, where $\vec{n}$ is the normal vector to the segment before advection and $\vec{d}$ the displacement of its center during advection. After advection, the cell is full if, $pr > 0$ ($C = 1$) (*figure 6*), and

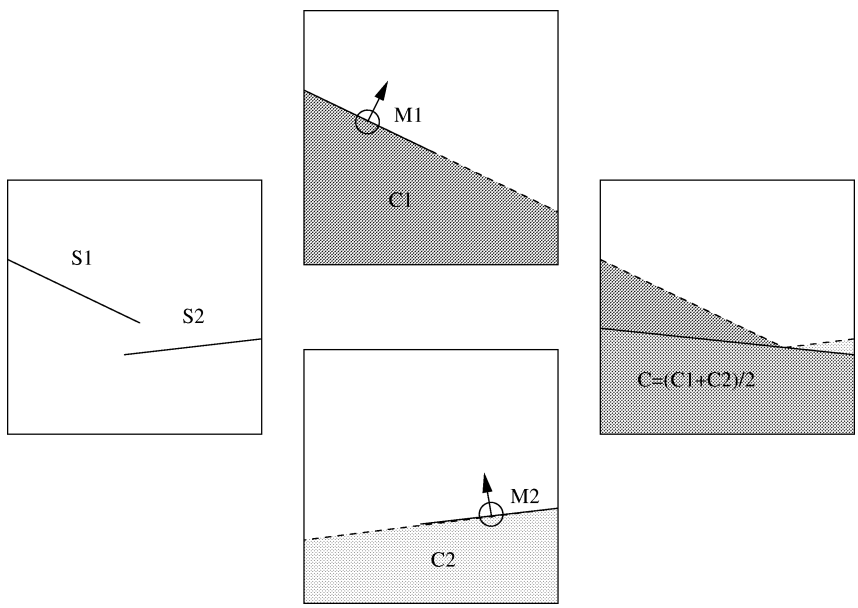

Figure 3. VOF calculation for a type A cell without reconnection.

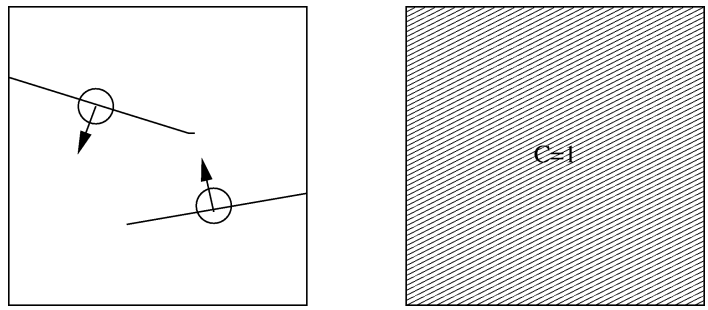

Figure 4. VOF calculation for a type A cell with reconnection.

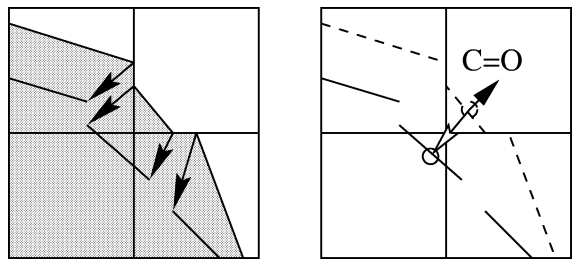

Figure 5. VOF calculation for a type B cell, $pr < 0$: $C = 0$. Left: black arrows are velocity vectors. Right: black and white arrows are normal and velocity vectors respectively.

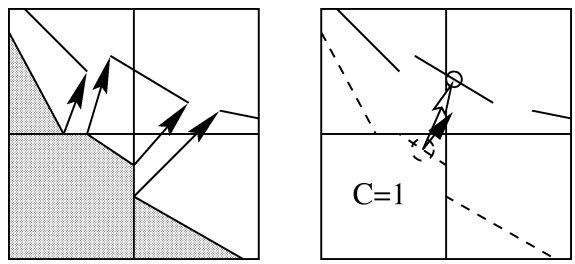

Figure 6. VOF calculation for a type B cell, $pr > 0$: $C = 1$. Left: black arrows are velocity vectors. Right: black and white arrows are normal and velocity vectors respectively.

empty if not ($C = 0$) (*figure 5*). One of the advantages of this method, compared to the former PLIC method, is to allow the use of larger time steps.

Thus, all the computations presented in this paper are performed with adaptive time-steps under the condition $\mathrm{CFL}_{vof} = 0.9$, where the CFL_{vof} criterion is defined as the maximun value of the ratio of the displacement of a particle at the interface during a time step to the maximun size of this cell. Note that earlier VOF methods were limited by the stability criterion $\mathrm{CFL}_{vof} < 0.5$ (Hirt and Nichols [10]; Li [12]; Abadie [14]).

2.4. Boundary conditions

In the pseudo-compressibility method, the boundary conditions are applied at each R-K step. Two types of boundaries are defined.

2.4.1. Open and solid boundaries

For solid boundaries, free-slip conditions are applied: $\vec{v}_n = 0$ and for open boundaries, stress-free conditions are applied:

$$\begin{aligned} \frac{\partial \vec{v}}{\partial n} &= 0, \\ p &= 0. \end{aligned} \tag{18}$$

2.4.2. Free surfaces and interfaces

A moving boundary separating air and water can be considered in two ways:

(i) If the effect of air is neglected, the boundary is a free surface. Thus the pressure at this boundary is constant and in fact equals to zero here. Numerically, at each R-K step, the velocity and the pseudo-density in the partial cells ($0 < C < 1$) of the grid are calculated from those of the neighboring non-empty cells ($C > 0$) by the following formula:

$$\vec{U}_{i,j}^{m+(q+1)/5} = \frac{\sum_{(k=i-1,i+1),(l=j-1,j+1)} C_{k,l} \vec{U}_{k,l}^{m+q/5}}{\sum_{(k=i-1,i+1),(l=j-1,j+1)} C_{k,l}}, \tag{19}$$

where $\vec{U}_{k,l}^{m+q/5}$ is the velocity in the cell (k, l) at R-K step q and $\vec{U}_{i,j}^{m+(q+1)/5}$ is the velocity in the cell (i, j) at R-K step $q + 1$.

The pressure in the partial cells is fixed equal to the atmospheric pressure taken here equal to zero.

(ii) If the coupling between air and water is taken into account, the boundary is an interface. In this case, we assume continuity of the pressure and velocity across the interface. Velocity and pseudo-density of the partial cells are calculated from the values of their neighbors by giving a larger weight to the cells containing the denser fluid. Numerically, the following formula is used:

$$\vec{U}_{i,j}^{m+(q+1)/5} = \frac{\sum_{(k=i-1,i+1),(l=j-1,j+1)} [(\rho_l - \rho_g) C_{k,l} + \rho_g] \vec{U}_{k,l}^{m+q/5}}{\sum_{(k=i-1,i+1),(l=j-1,j+1)} [(\rho_l - \rho_g) C_{k,l} + \rho_g]}, \tag{20}$$

where ρ_l and ρ_g are the real densities of water and air, respectively.

The pressure in the partial cells is supposed to be driven by the air pressure and is computed as follows:

$$p_{i,j}^{m+(q+1)/5} = \frac{\sum_{(k=i-1,i+1),(l=j-1,j+1)} (1 - C_{k,l}) p_{k,l}^{m+q/5}}{\sum_{(k=i-1,i+1),(l=j-1,j+1)} (1 - C_{k,l})}, \tag{21}$$

where $p_{k,l}^{m+q/5}$ is the pressure in the cell (k, l) at R-K step q and $p_{i,j}^{m+(q+1)/5}$ is the pressure in the cell (i, j) at R-K step $q + 1$.

2.5. Validation of the model

2.5.1. Kinematic validation: the disk advection

A disk was advected in an uniform, constant, horizontal velocity field, equivalent to one cell per second. The grid size was 200×100. Grid cells were square. The radius of the disk was 30 cells long. *Figure 7* displays the initial C field and the segments representing the interface. *Figure 8* shows the C field at $t = 90.15$ s after

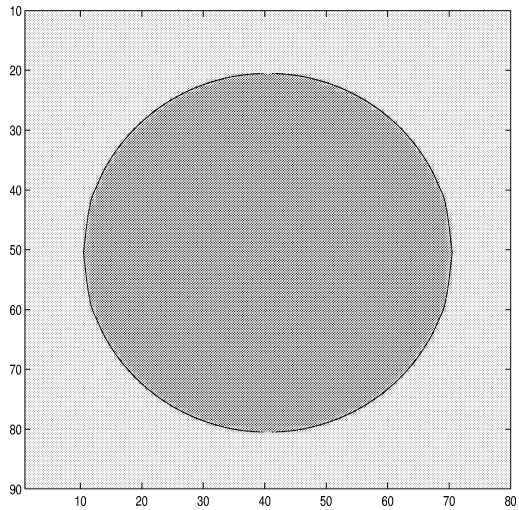

Figure 7. Initial circle discretization at $t = 0.0$ s using the PLIC method. The black line is a collection of segments as the shade variation corresponds to the C values.

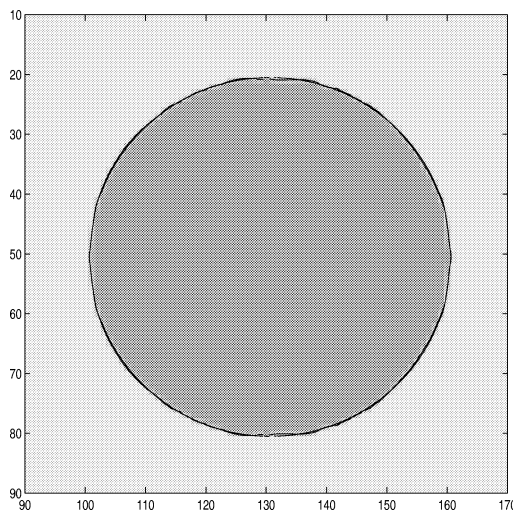

Figure 8. Circle advection by SL-VOF at $t = 90.15$ s.

advection by SL-VOF. The segments deduced from the C field and the segments deduced from the initial ones by longitudinal translation of one times the simulation time (90.15 s) are shown in *figure 8*. Very small differences are observed.

2.5.2. Dynamic validation: Rayleigh–Taylor instability

In order to test the capability of our model to simulate two-phase flows with strongly non-linear interfaces, we considered the classical Rayleigh–Taylor instability problem between two fluids. The density of the lower fluid is 0.5 and that of the upper fluid is 1.

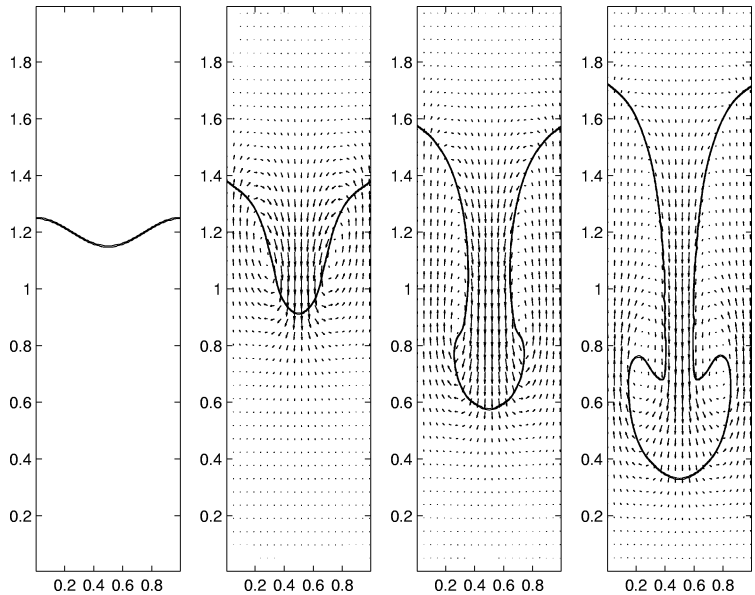

Figure 9. Rayleigh–Taylor instability at $t = 0.0000, 0.7180, 1.1627, 1.4700$ (s).

The computational domain is 2 m high by 1 m wide, and the mesh size is 128×57. The initial perturbation of the interface is sinusoidal with a 0.05 m amplitude and 1 m wavelength. The gravitational acceleration is taken equal to 9.81 ms^{-2} and the dynamic viscosity of both fluids is 0.003 $\mathrm{kgm}^{-1}\mathrm{s}^{-1}$.

The iso-C contours for $C = 0.3$ and $C = 0.7$ at times $0, 0.7180, 1.1627, 1.4700$ (s) are shown in *figure 9*. The velocity field is also described and the maximum velocity value is equal to 2.74 ms^{-1}. It clearly appears that the interface is quite smooth and that the fluid motion problem presents a vertical symmetry. The fact that the two iso-C contours keep being close together shows that the interface tracking method does not diffuse the interface, even for a strongly disturbed flow.

3. Application to solitary wave shoaling

The method was applied to the propagation of a solitary wave over a sloping bottom. This type of wave is very convenient for numerical simulation since the boundary conditions are very simple: the velocity field far from the crest is zero. The computations are divided into two steps. The shoaling phase of the wave is simulated without considering the air flow, so that the water surface is a free surface. In this case free slip conditions are applied on all the fixed boundaries. After the breaking instant, the Navier–Stokes equations are solved in both fluids and the boundary between the two fluids is treated as an interface (see section 2.4.2). On the bottom (flat + slope) and the left side of the computational box, free slip conditions are imposed, whereas the top and the right vertical sides of the box are considered as air stress-free boundaries. The initial pressure, velocity field and interface shape are those of an exact solitary wave for the potential flow problem, propagating over a flat bottom. The initial free surface shape and the values of the velocity potential at the free surface are given by Tanaka's algorithm (Tanaka [23]). From those data, the initial pressure and velocity field are computed using the Grilli et al.'s [4] BIEM method (*figure 10*). The grid size is 80×2500, and the mesh is curvilinear: the bottom consists of two straight segments belonging to the same domain boundary. The cells are of varying

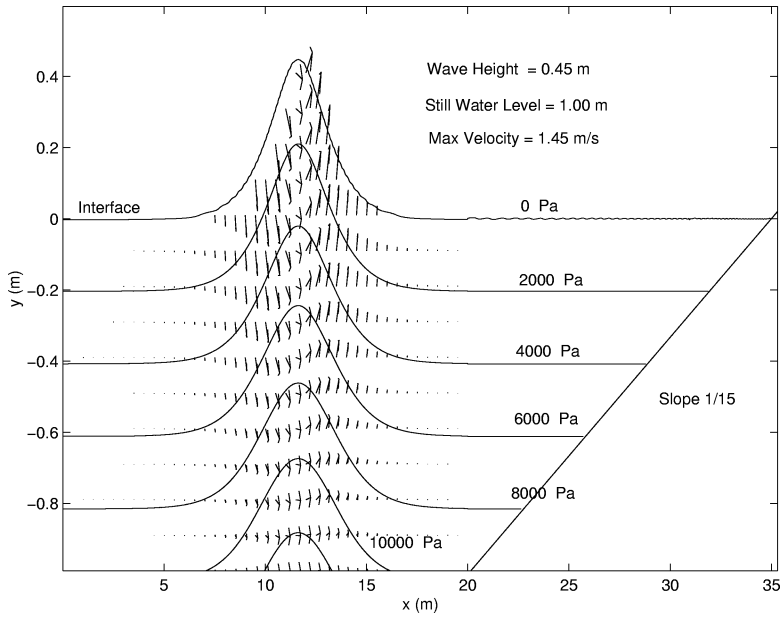

Figure 10. Initial condition deduced from Tanaka's algorithm using a BIEM method.

dimensions (0.025 m $\times$ 0.02 m at the left side of the box and 0.0125 m $\times$ 0.01 m at the right vertical side of the box). Total computational time was 40 hours on a Dec-alpha 500 UNIX workstation.

3.1. Comparison with BIEM

The case of an exact solitary wave of 0.45 m height propagating over a horizontal bottom of constant depth of 1 m, at first, and then over a constant sloping bottom of 1/15 slope was studied (*figure 11*). The evolution of wave characteristics during propagation over the slope towards the beach (shoaling) is compared with BIEM results by Grilli et al. [4] which are known to be very accurate.

3.1.1. Wave phase velocity

Figure 12 shows the time evolution of the crest abscissa for both methods, during propagation over the flat bottom and then over the slope. A good agreement is observed showing the capability of the SL-VOF method in modelling solitary wave propagation on a sloping bottom.

Nevertheless, the solitary wave computed by the SL-VOF method propagates slightly slower than that predicted by the BIEM. This is due to the loss of the wave amplitude in the SL-VOF computations which is shown in *figure 11*.

3.1.2. Interface shape at breaking point

The shoaling phenomenon leads to an increase in wave height, steepness and profile asymmetry. This was established by many authors (for a review, see e.g. Grilli et al. [4]). These parameters are relevant for the validation of our method. In *figure 11*, the interface shapes computed by both methods for the same initial solitary wave are shown at the breaking point. The breaking point is defined as the instant, the interface slope is vertical for the first time during the propagation. The interface shapes are in close agreement,

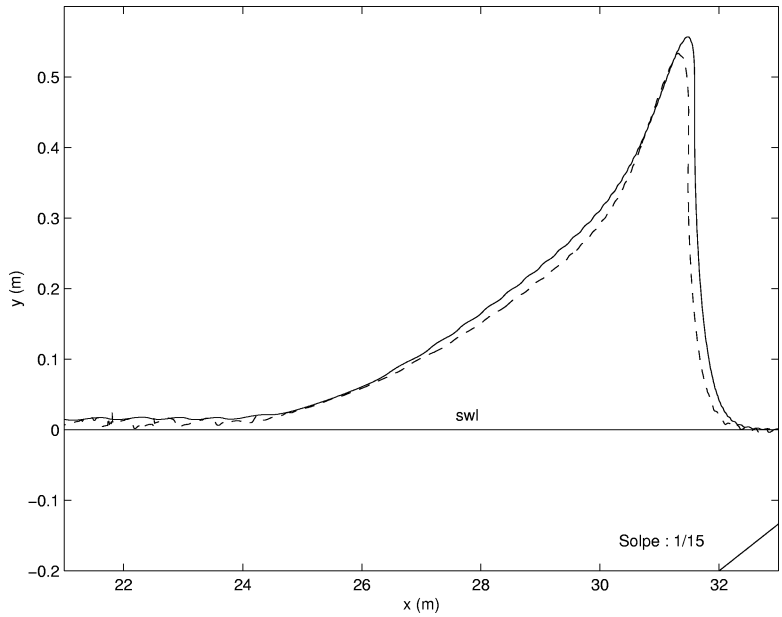

Figure 11. Interface shape at the breaking point. BIEM (-) at $t = 5.19$ s, SL-VOF (- -) at $t = 5.03$ s.

especially concerning their asymmetry, crest elevation and steepness. The small differences observed here can be explained by the energy loss in our computations due to the numerical diffusion.

3.1.3. Energy and volume conservation

The total volume loss in our simulations was computed relatively to the wave volume. The maximum error is of one percent from the beginning (*figure 10*) up to the breaking point (*figure 11*).

Concerning the energy conservation, we compared the energy flux through the vertical plane $X = 20.00$ m, obtained by both methods: the energy flux being here the variation of the energy contained in the domain defined by $X > 20.00$ m from the start up to the breaking point. The loss of the SL-VOF method compared to the BIEM is of 9.91 percent. This result is quite satisfactory since it corresponds to a 20 m long propagation.

3.1.4. Pressure field

Many surf-zone models assume that the pressure under waves is hydrostatic. *Figure 13* shows the iso-pressure contours computed by the code and those deduced from the interface shape under the hydrostatic assumption. A large difference between the two models is observed, especially when the wave steepness is very high, i.e. at the breaking point.

3.2. Breaking event: comparison with experiments

3.2.1. Experimental set-up and wave generation

The experiments were carried out at the Institut des Sciences de l'Ingénieur de Toulon et du Var (ISITV)'s wave tank. This glass-walled tank allows us to make video acquisition of the wave shape at different steps of its propagation. The dimension of the flume also shown in *figure 14* ensures the flow is two-dimensional.

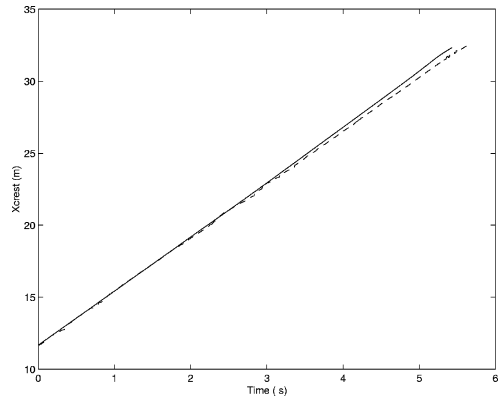

Figure 12. Crest abscissa versus time for each method. BIEM (-) , SL-VOF (- -).

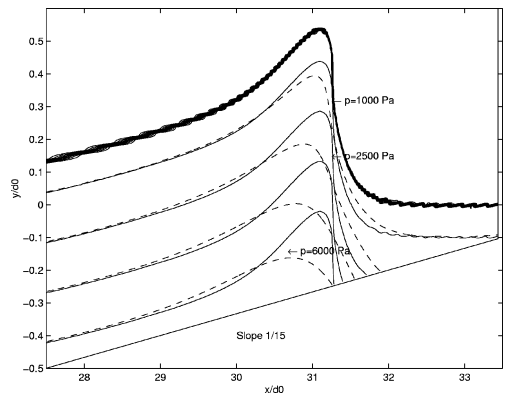

Figure 13. Pressure field at the breaking point: comparison with hydrostatic models. (—): pressure computed with SL-VOF; (-): hydrostatic pressure, twisted black lines represent C contours.

• Free surface shape sampling:

Interface shapes during the wave propagation and breaking were recorded using a S-VHS camera (24 frames per second), displayed on a large screen and digitized. The digitized images are then used to manually select points on the free-surface. A first free surface profile is thus defined. This profile is unfortunatly distorted by the screen print. The local distortion is corrected by doing a comparision of markers' positions on the flume transparent sides and on the digitized images. All these operations on the digitized images are made using Matlab post-processing routines, developed for this purpose.

• Wave generation:

The wave generation was made using a flat paddle (*figure 14*) moved by a rubber band released suddenly. Such a simple way to generate waves may produce unexpected wave shapes. In order to be sure that the generated waves were solitary waves, the following experimental procedure was used: the paddle motion was forced to have a forward displacement. The wave shape evolution was then recorded with the camera in the region of flat bottom (Control window in *figure 14*). To be considered a solitary wave, the wave had to satisfy the following criterion: the interface shape had to be close enough to the profile given by Tanaka's algorithm (Tanaka [23]) and the amplitude of the dispersive waves behind the main wave had to be negligible. *Figure 15* shows the results of the comparison between the exact interface shape of a solitary wave and the interface shape measured in the wave flume. The maximum error on the elevation with respect to depth is 2.5%. The experimental wave can thus be considered close enough to the exact solitary wave.

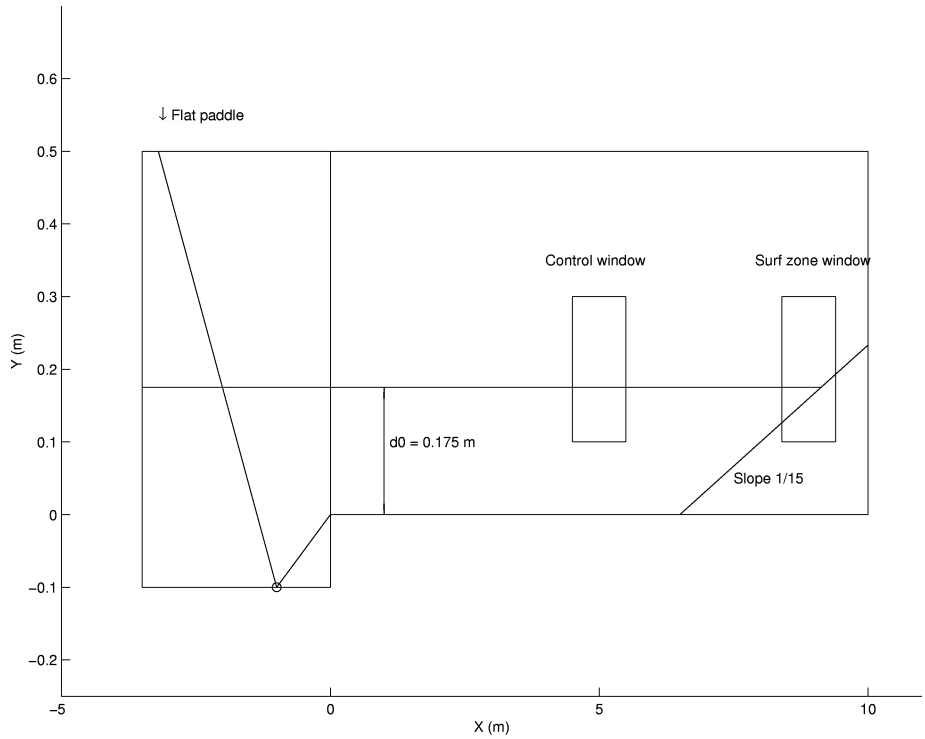

Figure 14. The ISITV wave flume.

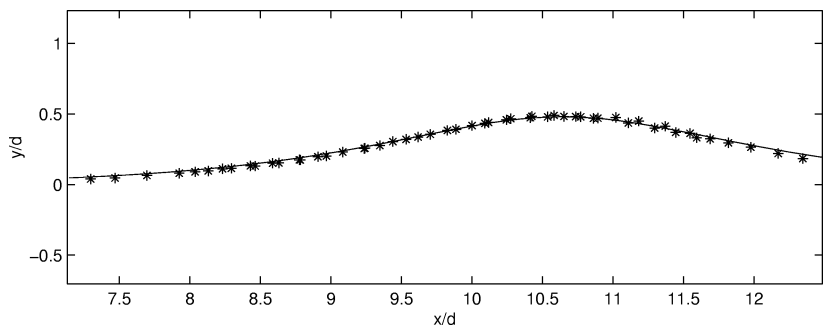

Figure 15. Comparison of the initial wave shape with exact solitary wave shape: (*) experiments, (–) exact Tanaka's profile.

3.2.2. Comparisons

Our computations were compared with experiments carried out for a 0.084 m height solitary wave propagating, first over a $d_0 = 0.175$ m constant water depth, and then shoaling over a 0.0773 slope. Computations were performed under the same conditions, as presented in section 3.1, except that the slope is now 0.0773 and the initial wave height is 0.48 m. *Figure 16* compare the interface shape evolution of the experiments and the SL-VOF method at $t_c = \frac{t-t_{ref}}{\sqrt{d_0}} = 0, 0.10, 0.19, 0.27, 0.38$ s for the experiments, and $t_c = 0, 0.11, 0.22, 0.32, 0.41$ (s) for the computations. The interface shape at $t - t_{ref} = 0$ s (first frame) shows, for the breaking area, a reasonable agreement between the SL-VOF model and the experiments. As shown in section 3.1, the present model gives a good estimate of the breaking abscissa. The fact that the experimental wave breaks at the same location as the computed wave suggests that both experimental and numerical initial waves are close enough for our study. The interface shapes after $t - t_{ref} = 0$ s are similar and prove the capability

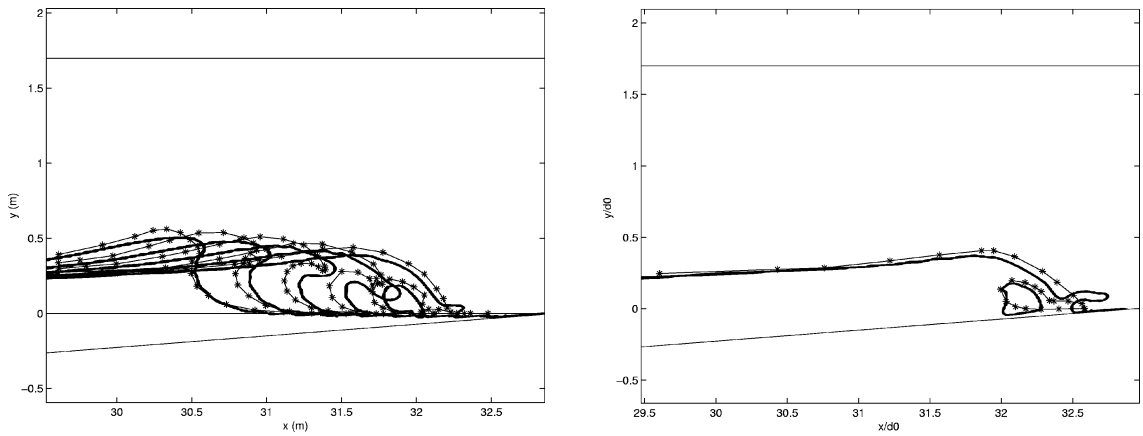

Figure 16. Interface shapes during breaking. Experiments (-*) SL-VOF(-).

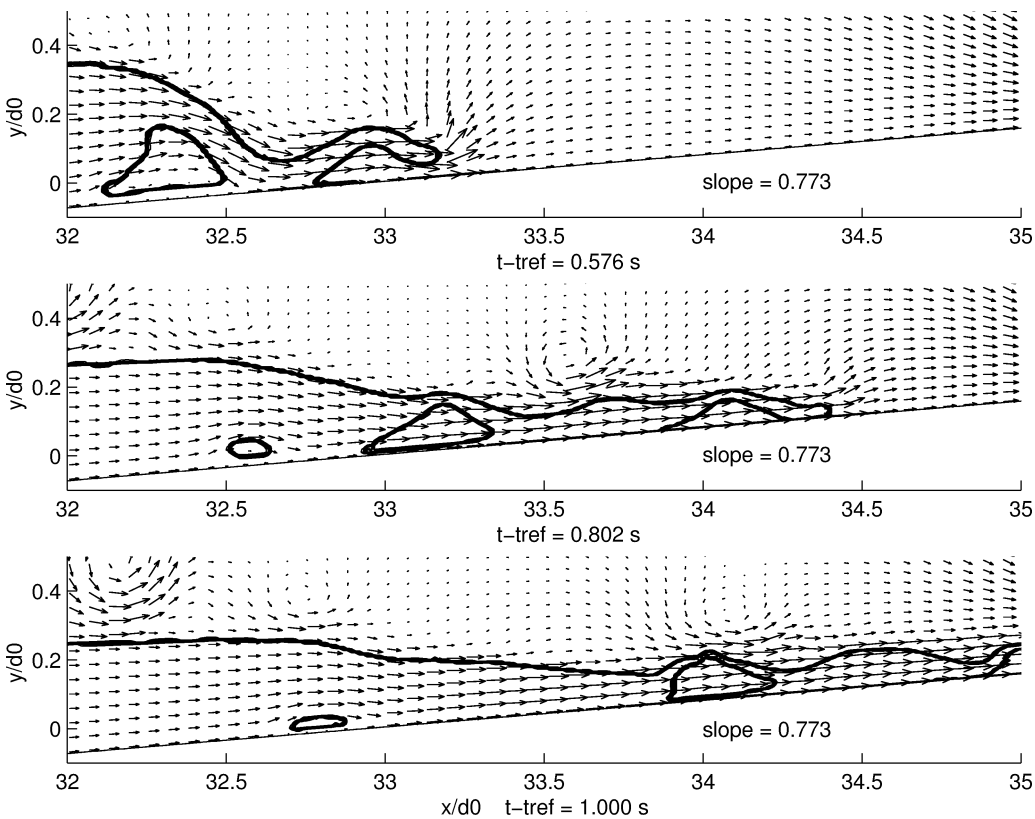

Figure 17. Post breaking stage: splash up generation and wave collapsing.

of our model to simulate the breaking phenomenon. The splash-up phenomenon mentioned by many authors (Abadie [14], Bonmarin [24], Chen et al. [15] and Peregrine [25]), appears clearly in *figure 16* (right part). Finally, *figure 17* describes the computed flow after breaking: the wave has disappeared and the flow looks like a current climbing up the slope: this is the swash phenomenon.

4. Conclusion

A numerical method able to simulate two-phase flows separated by highly non-linear interfaces has been developed. Simple test cases like advection of a circle and of a Rayleigh–Taylor instability have been successfully simulated. This confirms the capability of the method to accurately compute very complex flows. By applying our method to the case of a solitary wave shoaling over a plane slope, and after comparison with well-known BIEM results, we showed that our model (i) correctly simulates wave shoaling with only weak energy and volume losses, and (ii) describes rather well the interface shape, up to the breaking point. Finally, the interface deformation during breaking is compared with simple two-dimensional experiments and a good agreement is obtained. These results show that our model is able to simulate wave breaking due to spatial bottom variations including the description of the fine structure of the interface shape.

Results concerning the velocity and pressure fields exhibit a highly non-hydrostatic flow near breaking, which cannot be predicted by classical non-linear hydrostatic models. Furthermore, an accurate description of the velocity field during breaking is of great interest for nearshore modelling.

One weakness of our model is the large computational time required for the numerical simulations. In order to reduce this problem, studies concerning the coupling of our method with a BIEM method are being carried out. This will allow to focus on the breaking area, and to extend the method to periodic waves, or to study in detail the air-water interaction under windy conditions and during the post-breaking stage. Such computations would allow us to both reduce the cell size and the numerical diffusion pointed out in section 3.1.3. Finally, more realistic flow conditions would require fully three-dimensional computations, involving subscale model for turbulence and phase mixing.

Acknowledgments

The help and BIEM software and Tanaka's algorithm provided by Professor Stephan Grilli during his stay an invited professor in Université de Toulon et du Var (1999) are gratefully acknowledged. Financial support to Stephan Guignard by the Centre National de la Recherche Scientifique and Principia R and D (France) are also acknowledged. This research was supported by the French Program National d'Environnement Côtier and EC Mast III Program FANS (Contract MAS3-CT95-0037).

References

[1] Longuet-Higgins M.S., Cokelet E.D., The deformation of steep surface waves on water. II: A numerical method for computation, P. Roy. Soc. Lond. A Mat. 364 (1976) 1–28.
[2] Peregrine D.H., Computation of breaking waves, Water Wave Kinematics (1990) 475–490.
[3] Skyner D., A comparison of numerical prediction and experimental measurements of the internal kinematics of a deep water plunging breaker, J. Fluid. Mech. 315 (1996) 51–64.
[4] Grilli S.T., Svendsen I.A., Subramanya R., Breaking criterion and characteristics for solitary wave breaking on slopes, J. Waterw. Port C-ASCE 123 (3) (1997) 102–112.
[5] Yasuda T., Mutsuda H., Oya A., Tada A., Fukumoto T., A new type Breaker forming a Giant jet and its decaying propertise, Proc. XXIVth ICCE, 1996, pp. 704–718.
[6] Yasuda T., Mutsuda H., Mizutani N., Kinematic of overturning solitary waves and their relations to breaker types, Coast. Eng. 29 (1997) 317–346.
[7] Grilli S.T., Losada M.A., Martin F., Breaking criterion and characteristics for solitary wave breaking on slopes, J. Waterw. Port C-ASCE 120 (1) (1994) 74–92.
[8] Hyman J.M., Numerical method for tracking interfaces, Physica D (1984) 396–407.
[9] Monaghan J.J., Bicknell P.J., Humble R.J., Volcanoes, tsunamis and the demise of the minoans, Physica D 77 (1994) 217–228.
[10] Hirt C.W., Nichols B.D., Volume Of Fluid method for the dynamics of free boundaries, J. Comp. Phys. 39 (1981) 323–345.

[11] Lafaurie B., Nardone C., Scardovelli R., Zaleski S., Zanetti G., Modelling merging and fragmentation in multiphase flows with SURFER, J. Comp. Phys. 113 (1994) 134–147.
[12] Li J., Piecewise linear interface calculation, C.R. Acad. Sci. IIb 320 (1995) 391–396.
[13] Petit H.A., Tonjes P., Van Gent M.R.A., Van Den Bosch P., Numerical simulation and validation of plunging breaker using a 2D Navier Stokes Model, Proc. XXIVth ICCE, 1994, pp. 511–524.
[14] Abadie S., Modélisation numérique du déferlement plongeant par une méthode VOF, Thèse de doctorat, Université Bordeaux I, 1998.
[15] Chen G., Kharif C., Zaleski S., Li J., Two-dimensional Navier–Stokes simulation of breaking waves, Phys. Fluid 11 (1) (1999) 121–133.
[16] Lin P., Liu. P.L.F., Turbulence transport, vorticity dynamics, and solute mixing under plunging breaking waves in the surf zone, Jour. Geo. Res. 103 (C8) (1998) 15, 677–15, 694.
[17] Lin P., Liu. P.L.F., A numerical study of breaking waves in the surf zone, J. Fluid Mech. 359 (1998) 239–264.
[18] Viviand H., Analysis of pseudo compresibility systems for compressible and incompressible flows, Comput. Fluid Dyn. Rev. 29 (3) (1995) 650–655.
[19] Laget O., Dias F., Numerical simulation of capillary-gravity interfacial solitary waves, J. Fluid Mech. 349 (1997) 221–251.
[20] Laget O., Résolution des équations d'Euler pour les écoulements non-linéaires en presence d'une interface, Thèse de doctorat, Université de Nice-Sophia Antipolis, 1998.
[21] Vatsa V.N., Accurate numerical solution for transonic viscous flow over finite wings, J. Aircraft 24 (3) (1987) 377–385.
[22] Jameson A., Baker T.J., Solution of the Euler equation for complex configurations, AIAA Paper 83-1929, 1983.
[23] Tanaka M., The stability of solitary waves, Phys. Fluids 29 (3) (1986) 650–655.
[24] Bonmarin P., Geometric properties of deep water breaking waves, J. Fluid Mech. 209 (1989) 405–433.
[25] Peregrine D.H., Breaking waves on beaches, Ann. Rev. Fluid Mech. 15 (1983) 149–178.
[26] Schaffer H.A., Madsen P.A., Deigaard R., A Boussinesq model for waves breaking in shallow water, Coast. Eng. 20 (1993) 185–202.